Documentation Index
DI-2

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
INFORMATION PROCESSING CENTER

July 6, 1970

KEYWORD INDEX OF MULTICS PROGRAMMERS' MANUAL

This publication is a keyword index by content of the Multics Programmers' Manual for selected keywords. Appendix B includes the text from which the keywords were chosen. Appendix A is the program used to compile the index. This index will be updated periodically when necessitated by revisions to the manual.

)Index for Multics - 19 words beginning with)

"0" 4
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
remark: if the record quota of a directory is 0, its pages are charged to its parent directory
AMLcmd: entry_usage$clear - set entrypoint usage counts to 0
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0

"1" 9
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 1.1 describes the login and logout commands and system messages
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 1.6 describes punch-card input and control cards
remark: MPM RefData 2.1 describes local conventions for Multics users
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: random_$exponential - return random number from exponential distribution with mean 1

"10" 3
SSSsub: cv_bin_$dec - convert binary to character string, base 10
SSSsub: cv_dec_ - convert character string to binary, base 10
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code

"1024" 1
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand

"128" 1
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)

"2" 11
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: MPM RefData 2.1 describes local conventions for Multics users
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
AMLsub: roundb - round a number to next multiple of 2**n
AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
AMLsub: roundb$round36 - round a number to multiple of 36, round to multiple of 2**n

"200" 2
SSScmd: page_trace (pgt) - print out information about last 200 page faults
SSSsub: hcs_$get_page_trace - return information about last 200 page faults

"2741" 1
remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)

"3" 8
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
remark: MPM RefData 3.4 describes the linkage section and inter-segment references
remark: MPM RefData 3.5 describes magnetic tape formats

```
     remark: MPM RefData 3.7 describes punched card codes

"36" 2
     AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
     AMLsub: roundb$rounib36 - round a number to multiple of 36, round to multiple of 2**n

"360" 1
     AMLcmd: convert_360_fortran (s360f) - convert programs in the fortran language from 360 format

"37" 1
     remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)

"4" 3
     remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
     remark: MPM RefData 1.4 describes the input-output system and device and ourtar module types
     remark: MPM RefData 3.4 describes the linkage section and inter-segment references

"5" 2
     remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
     remark: MPM RefData 3.5 describes magnetic tape formats

"6" 1
     remark: MPM RefData 1.6 describes punch-card input and control cards

"7" 3
     remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
     remark: MPM RefData 2.7 describes the system standard search rules used by the linker
     remark: MPM RefData 3.7 describes punched card codes

"8" 3
     SSSsub: cv_bin_$oct - convert binary to character string, base 8
     SSSsub: cv_oct_ - convert character string to binary, base 8
     SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code

"9" 1
     remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)

">" 3
     remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry
     remark: an absolute path name begins with ">" and leads from the root directory to some entry
     remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
```

"ACL" 12
remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch
remark: each item on an ACL gives an access name and the access mode permitted
SSScmd: deleteacl (da) - delete item from an ACL
SSScmd: listacl (la) - print out ACL for a file
SSScmd: setacl (sa) - add item to ACL for a file
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL

"AML" 3
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM

"AMLcmd" 32
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
AMLcmd: adjust - set bit count on file branch
AMLcmd: adjust$block - set bit count on file branch, start from current length
AMLcmd: adjust$block_test - check bit count on file branch, start from current length
AMLcmd: adjust$test - check bit count on file branch
AMLcmd: adjust$test_block - check bit count on file branch, start from current length
AMLcmd: ascii_check - check all characters of a file for valid ASCII
AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII
AMLcmd: bcpl - compile a program in the bcpl language
AMLcmd: compare_ascii (cpa) - compare two character files
AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
AMLcmd: createSymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
AMLcmd: debug (db) - symbolic interactive debugging aid
AMLcmd: entry_usage - print out usage counts for entrypoints
AMLcmd: entry_usage$clear - set entrypoint usage counts to 0
AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language
AMLcmd: extract_archive - extract all component files from a CTSS archive file
AMLcmd: fs_chname - add, delete, or change entry names on a file
AMLcmd: ind - indent programs in the pl1 language
AMLcmd: lower_case - convert all characters in a file to lower case
AMLcmd: mail (ml) - send a file to another user's mailbox
AMLcmd: mail$unlock - unlock a locked mailbox
AMLcmd: pht - print horizontal tab
AMLcmd: pnl - print newline
AMLcmd: printsymtab (pst) - print symbol table for a pl1 program
AMLcmd: ps - print argument string
AMLcmd: qed - programmable context editor for character files
AMLcmd: return_to - return to a given stack frame
AMLcmd: return_to$repeat_call - return to given stack frame and repeat call
AMLcmd: runoff - format a character file for printing

"AMLsub" 38
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM
AMLsub: hcs_$block - block process until next wakeup

AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id
AMLsub: ipc_$block - block process until given event
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
AMLsub: ipc_$create_ev_chn - create an event channel
AMLsub: ipc_$cutoff - inhibit event channel for reading
AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)
AMLsub: ipc_$delete_ev_chn - delete an event channel
AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup
AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
AMLsub: ipc_$read_ev_chn - return event message from an event channel
AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels
AMLsub: random_$exponential - return random number from exponential distribution with mean 1
AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
AMLsub: random_$normal - return random number from normal distribution
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_seq - return array of random numbers from normal distribution
AMLsub: random_$random_get_seed - return current seed for random numbers
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$random_set_seed - set seed for random numbers
AMLsub: random_$uniform - return random number from uniform distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution
AMLsub: read_list_ - read in free-format variables from "user_input"
AMLsub: reverse_index - scan character string from right to left for given character
AMLsub: roundb - round a number to next multiple of 2**n
AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
AMLsub: roundb$rounib36 - round a number to multiple of 36, round to multiple of 2**n
AMLsub: whoami_ - return user's name and project as given at login
AMLsub: write_list_ - write variables on "user_output"

"ASCII" 5
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
AMLcmd: ascii_check - check all characters of a file for valid ASCII
AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII
AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format

"ATTN" 1
remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)

"Author" 1
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS

"aa_conv" 1
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format

"abort" 1
SSSsub: ios_$abort - abort outstanding transactions for an ioname

"absolute" 12
remark: an absolute path name begins with ">" and leads from the root directory to some entry
SSS_af: get_pathname (gpn) - return absolute path name for given reference name

```
"active_function_error" 1
     SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"

"add" 9
     SSScmd: addname (an) - add additional entry names to file branch
     AMLcmd: fs_chname - add, delete, or change entry names on a file
     SSScmd: setacl (sa) - add item to ACL for a file
     SSScmd: setcacl (sc) - add item to CACL for a directory
     SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
     SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
     SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
     SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
     SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry

"additional" 1
     SSScmd: addname (an) - add additional entry names to file branch

"addname" 1
     SSScmd: addname (an) - add additional entry names to file branch

"address" 3
     remark: a process has a two-dimensional address space consisting of segments
     remark: a "segment" is an addressable collection of words which has names and access attributes
     remark: the segments addressable by a process are grouped into rings of protection

"adjust" 5
     AMLcmd: adjust - set bit count on file branch
     AMLcmd: adjust$block - set bit count on file branch, start from current length
     AMLcmd: adjust$block_test - check bit count on file branch, start from current length
     AMLcmd: adjust$test - check bit count on file branch
     AMLcmd: adjust$test_block - check bit count on file branch, start from current length

"admin" 1
     SSScmd: who$admin - print out names, project id's, and process id's for users on system

"ahead" 1
     SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname

"aid" 2
     AMLcmd: debug (db) - symbolic interactive debugging aid
     SSScmd: probe - interactive debugging aid

"alloc_" 1
     SSSsub: alloc_ - allocate storage in given area, return pointer

"allocate" 3
     SSSsub: alloc_ - allocate storage in given area, return pointer
     SSSsub: freen_ - free allocated storage in an area
     SSSsub: hcs_$assign_linkage - allocate storage in linkage section

"allow" 1
     remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor

"another" 8
     remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
     AMLcmd: mail (ml) - send a file to another user's mailbox
     SSScmd: moveb (mv) - move a file to another directory
     SSSsub: hcs_$fs_move_file - move contents from one file branch to another
```

SSSsub: hcs_$fs_move_seg - move contents from one segment to another
SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSsub: ios_$detach - un-attach one ioname from another ioname
SSSsub: move_ - move a block of words from one place to another (given by pointers)

"antithetic" 4
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution

"append" 1
remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null

"append_branch" 1
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL

"append_branchx" 1
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL

"append_link" 1
SSSsub: hcs_$append_link - create a link entry in parent directory

"archive" 8
SSScmd: archive (ac) - combine several files into an archive file
SSScmd: archive_sort (as) - sort an archive file on component file names
AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments
AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
AMLcmd: extract_archive - extract all component files from a CTSS archive file
SSScmd: reorder_archive - change order of component files in an archive file

"archive_sort" 1
SSScmd: archive_sort (as) - sort an archive file on component file names

"area" 4
SSSsub: alloc_ - allocate storage in given area, return pointer
SSSsub: area_ - initialize an area
SSSsub: area_$redef - change the length of an area
SSSsub: freen_ - free allocated storage in an area

"area_" 2
SSSsub: area_ - initialize an area
SSSsub: area_$redef - change the length of an area

"arg_count" 1
SSSsub: cu_$arg_count - return number of arguments of calling procedure

"arg_list_ptr" 1
SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list

"arg_ptr" 1
SSSsub: cu_$arg_ptr - return pointer to character-string argument

"argument" 9
remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
SSScmd: iocall - call "ios_" with given arguments

AMLcmd: ps - print argument string
SSSsub: cu_$arg_count - return number of arguments of calling procedure
SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
SSSsub: cu_$arg_ptr - return pointer to character-string argument
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments

"array" 5
AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
AMLsub: random_$normal_seq - return array of random numbers from normal distribution
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution

"ascii_check" 2
AMLcmd: ascii_check - check all characters of a file for valid ASCII
AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII

"assembly" 1
AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language

"assign" 2
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: hcs_$unassign - request GIM to un-assign a device channel

"assign_linkage" 1
SSSsub: hcs_$assign_linkage - allocate storage in linkage section

"associate" 6
remark: every known segment in a process is associated with some file or directory branch
remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSsub: ios_$order - issue request to outer module associated with an ioname
SSSsub: ios_$seek - set reference pointers associated with an ioname
SSSsub: ios_$tell - return value of reference pointer associated with an ioname

"asynchronous" 2
SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname

"attach" 8
SSScmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console
SSScmd: print_attach_table (pat) - print ioname attachments for process
SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSsub: ios_$changemode - change mode of attachment of an ioname
SSSsub: ios_$detach - un-attach one ioname from another ioname
SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname

"attribute" 2
remark: a "segment" is an addressable collection of words which has names and access attributes
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link

"ba" 1
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments

"base" 7
SSSsub: cv_bin_ - convert binary to character string, any base
SSSsub: cv_bin_$dec - convert binary to character string, base 10
SSSsub: cv_bin_$oct - convert binary to character string, base 8
SSSsub: cv_dec_ - convert character string to binary, base 10
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSsub: cv_oct_ - convert character string to binary, base 8
SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code

"basic" 2
SSScmd: basic - compile and execute a program in the basic language
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language

"bcpl" 1
AMLcmd: bcpl - compile a program in the bcpl language

"begins" 2
remark: an absolute path name begins with ">" and leads from the root directory to some entry
remark: a relative path name is relative to the current working directory; it begins with "<" or a name

"between" 1
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory

"binary" 8
SSSsub: cv_bin_ - convert binary to character string, any base
SSSsub: cv_bin_$dec - convert binary to character string, base 10
SSSsub: cv_bin_$oct - convert binary to character string, base 8
SSSsub: cv_dec_ - convert character string to binary, base 10
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSsub: cv_oct_ - convert character string to binary, base 8
SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname

"bindarchive" 1
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments

"bit" 13
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
AMLcmd: adjust - set bit count on file branch
AMLcmd: adjust$block - set bit count on file branch, start from current length
AMLcmd: adjust$block_test - check bit count on file branch, start from current length
AMLcmd: adjust$test - check bit count on file branch
AMLcmd: adjust$test_block - check bit count on file branch, start from current length
SSScmd: set_bit_count (sbc) - set bit count for a file branch
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$status_minf - return bit count and type for a branch or link
SSSsub: hcs_$status_mins - return bit count and type for a segment
SSSsub: unique_bits_ - return unique bit string
SSSsub: unique_chars_ - convert bit string into unique character string

"block" 12
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them

AMLcmd: adjust$block - set bit count on file branch, start from current length
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds
SSSsub: acm_$sleep_interval - block process for time interval in microseconds
SSSsub: acm_$sleep_time - block process until time given in microseconds
AMLsub: hcs_$block - block process until next wakeup
AMLsub: ipc_$block - block process until given event
AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels
SSSsub: move_ - move a block of words from one place to another (given by pointers)

"block_test" 1
AMLcmd: adjust$block_test - check bit count on file branch, start from current length

"both" 1
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic

"bound" 1
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments

"brackets" 1
SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST

"branch" 41
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
remark: every known segment in a process is associated with some file or directory branch
remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
SSScmd: addname (an) - add additional entry names to file branch
AMLcmd: adjust - set bit count on file branch
AMLcmd: adjust$block - set bit count on file branch, start from current length
AMLcmd: adjust$block_test - check bit count on file branch, start from current length
AMLcmd: adjust$test - check bit count on file branch
AMLcmd: adjust$test_block - check bit count on file branch, start from current length
SSScmd: create (cr) - create a file branch
SSScmd: createdir (cd) - create a directory branch
SSScmd: deletedir (dd) - delete a directory branch and all files contained in it
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
SSScmd: deletename (dn) - delete entry name from a file branch
SSScmd: set_bit_count (sbc) - set bit count for a file branch
SSScmd: status (st) - print out status information from directory entry for a branch
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$fs_move_file - move contents from one file branch to another
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry

SSSsub: hcs_$set_copysw - set copy switch in branch entry.
SSSsub: hcs_$set_relatesw - set relate switch in branch entry
SSSsub: hcs_$status - return information from directory entry about a branch or link
SSSsub: hcs_$status_long - return all information from directory entry about a branch or link
SSSsub: hcs_$status_minf - return bit count and type for a branch or link
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
SSSsub: hcs_$truncate_file - truncate file branch from a given offset

"break" 4
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: parse_file_$parse_file_set_break - declare break characters to parse_file_
SSSsub: parse_file_$parse_file_unset_break - undeclare break characters for parse_file_

"brought" 1
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand

"bsys" 1
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language

"byte" 1
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)

"CACL" 9
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
SSScmd: deletecacl (dc) - delete item from a CACL
SSScmd: listcacl (lc) - print out CACL for a directory
SSScmd: setcacl (sc) - add item to CACL for a directory
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch

"CPU" 2
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used

"CTSS" 3
AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
AMLcmd: extract_archive - extract all component files from a CTSS archive file

"c360f" 1
AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format

"calc" 1
SSScmd: calc - desk calculator with memory for functions and expressions

"calculator" 2
SSScmd: calc - desk calculator with memory for functions and expressions
SSScmd: decam - desk calculator with memory

"calendar" 2
SSSsub: clock_ - return calendar clock time in microseconds
SSSsub: date_time_ - convert calendar clock time to character string date and time

"call" 27
remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
remark: users make calls to the input-output switch "ios_" to read and write on ionames
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
SSScmd: iocall - call "ios_" with given arguments
AMLcmd: return_to$repeat_call - return to given stack frame and repeat call
SSSsub: cu_$arg_count - return number of arguments of calling procedure
SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
SSSsub: cu_$cl - call current listener to re-enter command level
SSSsub: cu_$cp - call current command processor to execute command line
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame
SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments
SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame

AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels
SSSsub: signal_ - signal the occurrence of a condition, call its handler

"canonical" 1
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"card" 2
remark: MPM RefData 1.6 describes punch-card input and control cards
remark: MPM RefData 3.7 describes punched card codes

"carefully" 1
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines

"case" 1
AMLcmd: lower_case - convert all characters in a file to lower case

"cd" 1
SSScmd: createdir (cd) - create a directory branch

"change" 10
SSScmd: changewdir (cwd) - change current working directory
AMLcmd: fs_chname - add, delete, or change entry names on a file
SSScmd: rename (rn) - change an entry name for a file
SSScmd: reorder_archive - change order of component files in an archive file
SSSsub: area_$redef - change the length of an area
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$fs_search_set_wdir - change current working directory
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: ios_$changemode - change mode of attachment of an ioname

"changemode" 1
SSSsub: ios_$changemode - change mode of attachment of an ioname

"changewdir" 1
SSScmd: changewdir (cwd) - change current working directory

"channel" 29
remark: an "ioname" is the name of an input-output stream or device channel
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
remark: an event channel is used to receive inter-process messages and wakeups
remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds
SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: hcs_$get_status - return GIM status for device channel
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
SSSsub: hcs_$unassign - request GIM to un-assign a device channel
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2

```
     AMLsub: ipc_$create_ev_chn - create an event channel
     AMLsub: ipc_$cutoff - inhibit event channel for reading
     AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
     AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)
     AMLsub: ipc_$delete_ev_chn - delete an event channel
     AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup
     AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
     AMLsub: ipc_$read_ev_chn - return event message from an event channel
     AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)
     AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
     AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
     AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels

"character" 32
     remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
     remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
     remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
     AMLcmd: ascii_check - check all characters of a file for valid ASCII
     AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII
     AMLcmd: compare_ascii (cpa) - compare two character files
     SSScmd: edm - context editor for character files
     SSScmd: iomode - set character conversion mode for an ioname
     AMLcmd: lower_case - convert all characters in a file to lower case
     SSScmd: print (pr) - print a character file
     AMLcmd: qed - programmable context editor for character files
     AMLcmd: runoff - format a character file for printing
     SSSsub: com_err_$check_ts_errcode_ - return character string for system error code
     SSSsub: cu_$arg_ptr - return pointer to character-string argument
     SSSsub: cv_bin_ - convert binary to character string, any base
     SSSsub: cv_bin_$dec - convert binary to character string, base 10
     SSSsub: cv_bin_$oct - convert binary to character string, base 8
     SSSsub: cv_dec_ - convert character string to binary, base 10
     SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
     SSSsub: cv_oct_ - convert character string to binary, base 8
     SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
     SSSsub: date_time_ - convert calendar clock time to character string date and time
     SSSsub: date_time_$fstime - convert file system time to character string date and time
     SSSsub: ioa_$rs - format a message and return character string
     SSSsub: ioa_$rsnnl - format a message and return character string, no newline
     SSSsub: parse_file_ - scan character file and return indices of next break or symbol
     SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
     SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
     SSSsub: parse_file_$parse_file_set_break - declare break characters to parse_file_
     SSSsub: parse_file_$parse_file_unset_break - undeclare break characters for parse_file_
     AMLsub: reverse_index - scan character string from right to left for given character
     SSSsub: unique_chars_ - convert bit string into unique character string

"charged" 2
     remark: a directory may have a record quota, which limits the number of pages which may be charged to it
     remark: if the record quota of a directory is 0, its pages are charged to its parent directory

"check" 6
     AMLcmd: adjust$block_test - check bit count on file branch, start from current length
     AMLcmd: adjust$test - check bit count on file branch
     AMLcmd: adjust$test_block - check bit count on file branch, start from current length
     AMLcmd: ascii_check - check all characters of a file for valid ASCII
     AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII
     SSSsub: check_star_ - check whether path name contains a star
```

```
     SSScmd: login - identify user by name and project to system control and create user process
     SSScmd: new_proc - request user control to create a new user process
     SSSsub: get_group_id_ - return user's access control name
     SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star

"controls" 1
     remark: the GIOC interface module (GIM) controls device channels by means of DCW lists

"convention" 2
     remark: MPM RefData 2.1 describes local conventions for Multics users
     remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"conversion" 1
     SSScmd: iomode - set character conversion mode for an ioname

"convert" 13
     AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
     AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
     AMLcmd: lower_case - convert all characters in a file to lower case
     SSSsub: cv_bin_ - convert binary to character string, any base
     SSSsub: cv_bin_$dec - convert binary to character string, base 10
     SSSsub: cv_bin_$oct - convert binary to character string, base 8
     SSSsub: cv_dec_ - convert character string to binary, base 10
     SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
     SSSsub: cv_oct_ - convert character string to binary, base 8
     SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
     SSSsub: date_time_ - convert calendar clock time to character string date and time
     SSSsub: date_time_$fstime - convert file system time to character string date and time
     SSSsub: unique_chars_ - convert bit string into unique character string

"convert_360_fortran" 1
     AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format

"copy" 2
     SSScmd: copy (cp) - copy a file
     SSSsub: hcs_$set_copysw - set copy switch in branch entry

"core" 1
     remark: segments are divided into 1024-word pages which are brought into core by the file system on demand

"count" 13
     AMLcmd: adjust - set bit count on file branch
     AMLcmd: adjust$block - set bit count on file branch, start from current length
     AMLcmd: adjust$block_test - check bit count on file branch, start from current length
     AMLcmd: adjust$test - check bit count on file branch
     AMLcmd: adjust$test_block - check bit count on file branch, start from current length
     AMLcmd: entry_usage - print out usage counts for entrypoints
     AMLcmd: entry_usage$clear - set entrypoint usage counts to 0
     SSScmd: listotals (lt) - print out entry counts and use for a directory
     SSScmd: set_bit_count (sbc) - set bit count for a file branch
     SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
     SSSsub: hcs_$set_bc - set bit count in file branch entry
     SSSsub: hcs_$status_minf - return bit count and type for a branch or link
     SSSsub: hcs_$status_mins - return bit count and type for a segment

"cp" 2
     SSScmd: copy (cp) - copy a file
     SSSsub: cu_$cp - call current command processor to execute command line
```

```
"cpa" 1
     AMLcmd: compare_ascii (cpa) - compare two character files

"cr" 1
     SSScmd: create (cr) - create a file branch

"crawl_out_data" 1
     SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0

"create" 12
     SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments
     SSScmd: create (cr) - create a file branch
     SSScmd: createdir (cd) - create a directory branch
     AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
     SSScmd: link (lk) - create a link entry
     SSScmd: login - identify user by name and project to system control and create user process
     SSScmd: new_proc - request user control to create a new user process
     SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
     SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
     SSSsub: hcs_$append_link - create a link entry in parent directory
     SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
     AMLsub: ipc_$create_ev_chn - create an event channel

"create_ev_chn" 1
     AMLsub: ipc_$create_ev_chn - create an event channel

"createdir" 1
     SSScmd: createdir (cd) - create a directory branch

"createsymtab" 1
     AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")

"cst" 1
     AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")

"ctss_aarchv" 2
     AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
     AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format

"cu_" 15
     SSSsub: cu_$arg_count - return number of arguments of calling procedure
     SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
     SSSsub: cu_$arg_ptr - return pointer to character-string argument
     SSSsub: cu_$cl - call current listener to re-enter command level
     SSSsub: cu_$cp - call current command processor to execute command line
     SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
     SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame
     SSSsub: cu_$level_get - return current validation level
     SSSsub: cu_$level_set - set validation level
     SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments
     SSSsub: cu_$set_cl - define current listener
     SSSsub: cu_$set_cp - define current command processor
     SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
     SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
     SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame

"cumulative" 2
     SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
     SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used
```

```
"current" 25
     remark: a relative path name is relative to the current working directory; it begins with "<" or a name
     AMLcmd: adjust$block - set bit count on file branch, start from current length
     AMLcmd: adjust$block_test - check bit count on file branch, start from current length
     AMLcmd: adjust$test_block - check bit count on file branch, start from current length
     SSScmd: changewdir (cwd) - change current working directory
     SSScmd: get_com_line (gcl) - print out current expanded command line length
     SSScmd: getquota (gq) - print out record quota and current use for given directory
     SSScmd: how_many_users (hmu) - print out current number of users and maximum number
     SSScmd: printwdir (pwd) - print absolute path name of current working directory
     SSS_af: wd - return absolute path name of current working directory
     SSSsub: cu_$cl - call current listener to re-enter command level
     SSSsub: cu_$cp - call current command processor to execute command line
     SSSsub: cu_$level_get - return current validation level
     SSSsub: cu_$set_cl - define current listener
     SSSsub: cu_$set_cp - define current command processor
     SSSsub: get_ring_ - return the current ring number
     SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
     SSSsub: hcs_$fs_search_set_wdir - change current working directory
     SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
     SSSsub: ios_$getdelim - return current read delimiters for ioname
     SSSsub: ios_$getsize - return current element size for an ioname
     SSSsub: ios_$setdelim - set current read delimiters for an ioname
     SSSsub: ios_$setsize - set current element size for an ioname
     SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
     AMLsub: random_$random_get_seed - return current seed for random numbers

"cutoff" 2
     AMLsub: ipc_$cutoff - inhibit event channel for reading
     AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)

"cv_bin_" 3
     SSSsub: cv_bin_ - convert binary to character string, any base
     SSSsub: cv_bin_$dec - convert binary to character string, base 10
     SSSsub: cv_bin_$oct - convert binary to character string, base 8

"cv_dec_" 1
     SSSsub: cv_dec_ - convert character string to binary, base 10

"cv_dec_check_" 1
     SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code

"cv_oct_" 1
     SSSsub: cv_oct_ - convert character string to binary, base 8

"cv_oct_check_" 1
     SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code

"cwd" 1
     SSScmd: changewdir (cwd) - change current working directory
```

Index for Multics - 49 words beginning with d

"DCW" 7
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
SSSsub: hcs_$get_cur_status - return GIM position in DCW list
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list

"da" 1
SSScmd: deleteacl (da) - delete item from an ACL

"data" 2
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
SSSsub: ti_$findata - turn scratch segment into data segment

"date" 2
SSSsub: date_time_ - convert calendar clock time to character string date and time
SSSsub: date_time_$fstime - convert file system time to character string date and time

"date_time_" 2
SSSsub: date_time_ - convert calendar clock time to character string date and time
SSSsub: date_time_$fstime - convert file system time to character string date and time

"db" 1
AMLcmd: debug (db) - symbolic interactive debugging aid

"dc" 1
SSScmd: deletecacl (dc) - delete item from a CACL

"dd" 1
SSScmd: deletedir (dd) - delete a directory branch and all files contained in it

"debug" 3
AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: debug (db) - symbolic interactive debugging aid
SSScmd: probe - interactive debugging aid

"dec" 1
SSSsub: cv_bin_$dec - convert binary to character string, base 10

"decam" 1
SSScmd: decam - desk calculator with memory

"decl_ev_call_chn" 1
AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type

"decl_ev_wait_chn" 1
AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)

"declare" 1
SSSsub: parse_file_$parse_file_set_break - declare break characters to parse_file_

"decode_object_" 1
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment

"default" 2
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)

"define" 2
SSSsub: cu_$set_cl - define current listener
SSSsub: cu_$set_cp - define current command processor

"del_dir_tree" 1
SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory

"delayed" 4
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch
AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language
SSScmd: pl1d: request delayed compile of programs in the pl1 language

"delentry_file" 1
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory

"delentry_seg" 1
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory

"delete" 17
SSScmd: delete (dl) - delete a file
SSScmd: deleteacl (da) - delete item from an ACL
SSScmd: deletecacl (dc) - delete item from a CACL
SSScmd: deletedir (dd) - delete a directory branch and all files contained in it
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
SSScmd: deletename (dn) - delete entry name from a file branch
AMLcmd: fs_chname - add, delete, or change entry names on a file
SSScmd: unlink (ul) - delete a link entry from a directory
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname
AMLsub: ipc_$delete_ev_chn - delete an event channel

"delete_ev_chn" 1
AMLsub: ipc_$delete_ev_chn - delete an event channel

"deleteacl" 1
SSScmd: deleteacl (da) - delete item from an ACL

"deletecacl" 1
SSScmd: deletecacl (dc) - delete item from a CACL

"deletedir" 1
SSScmd: deletedir (dd) - delete a directory branch and all files contained in it

"deleteforce" 1
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary

"deletename" 1
SSScmd: deletename (dn) - delete entry name from a file branch

"delimiters" 2
SSSsub: ios_$getdelim - return current read delimiters for ioname
SSSsub: ios_$setdelim - set current read delimiters for an ioname

"demand" 1
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand

"describe" 9
remark: MPM RefData 1.1 describes the login and logout commands and system messages
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 1.6 describes punch-card input and control cards
remark: MPM RefData 2.1 describes local conventions for Multics users
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
remark: MPM RefData 3.4 describes the linkage section and inter-segment referances
remark: MPM RefData 3.5 describes magnetic tape formats
remark: MPM RefData 3.7 describes punched card codes

"desk" 2
SSScmd: calc - desk calculator with memory for functions and expressions
SSScmd: decam - desk calculator with memory

"destruction" 1
SSScmd: logout - request process destruction and session termination

"detach" 1
SSSsub: ios_$detach - un-attach one ioname from another ioname

"device" 10
remark: an "ioname" is the name of an input-output stream or device channel
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: hcs_$get_status - return GIM status for device channel
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
SSSsub: hcs_$unassign - request GIM to un-assign a device channel

"df" 1
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary

"dimension" 1
remark: a process has a two-dimensional address space consisting of segments

"directory" 46
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
remark: every known segment in a process is associated with some file or directory branch
remark: each process has a process directory, where per-process segments are kept
remark: an absolute path name begins with ">" and leads from the root directory to some entry
remark: a relative path name is relative to the current working directory; it begins with "<" or a name
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
remark: a directory may have a record quota, which limits the number of pages which may be charged to it
remark: if the record quota of a directory is 0, its pages are charged to its parent directory
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
SSScmd: changewdir (cwd) - change current working directory

```
     SSScmd: createdir (cd) - create a directory branch
     SSScmd: deletedir (dd) - delete a directory branch and all files contained in it
     SSScmd: getquota (gq) - print out record quota and current use for given directory
     SSScmd: global (gb) - execute a command in every directory inferior to a given directory
     SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
     SSScmd: listcacl (lc) - print out CACL for a directory
     SSScmd: listnames (ln) - print out entry names for files in a directory
     SSScmd: listotals (lt) - print out entry counts and use for a directory
     SSScmd: moveb (mv) - move a file to another directory
     SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
     SSS_af: pd - return absolute path name of process directory
     SSScmd: printhomedir (phd) - print absolute path name of home directory
     SSScmd: printwdir (pwd) - print absolute path name of current working directory
     SSScmd: setcacl (sc) - add item to CACL for a directory
     SSScmd: sethomedir (shd) - set path name of home directory
     SSScmd: status (st) - print out status information from directory entry for a branch
     SSScmd: unlink (ul) - delete a link entry from a directory
     SSS_af: wd - return absolute path name of current working directory
     SSSsub: get_pdir_ - return absolute path name of process directory
     SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
     SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
     SSSsub: hcs_$append_link - create a link entry in parent directory
     SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory
     SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
     SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
     SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
     SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
     SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
     SSSsub: hcs_$fs_search_set_wdir - change current working directory
     SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
     SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
     SSSsub: hcs_$star_ - search a directory for entry names matching template
     SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
     SSSsub: hcs_$status - return information from directory entry about a branch or link
     SSSsub: hcs_$status_long - return all information from directory entry about a branch or link

"distribution" 10
     AMLsub: random_$exponential - return random number from exponential distribution with mean 1
     AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
     AMLsub: random_$normal - return random number from normal distribution
     AMLsub: random_$normal_ant - return random number from antithetic normal distribution
     AMLsub: random_$normal_seq - return array of random numbers from normal distribution
     AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
     AMLsub: random_$uniform - return random number from uniform distribution
     AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
     AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
     AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution

"divide" 2
     AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
     AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n

"divided" 1
     remark: segments are divided into 1024-word pages which are brought into core by the file system on demand

"dl" 1
     SSScmd: delete (dl) - delete a file

"dn" 1
```

SSScmd: deletename (dn) - delete entry name from a file branch

"dp" 1
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer

"dpn" 1
SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch

"dprint" 1
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer

"dpunch" 1
SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch

"drain" 1
AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup

"drain_chn" 1
AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup

```
"each" 5
     remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
     remark: each process has a process directory, where per-process segments are kept
     remark: each item on an ACL gives an access name and the access mode permitted
     remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
     remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure

"ec" 1
     SSScmd: exec_com (ec) - execute a series of command lines contained in a file

"editor" 3
     SSScmd: bsys - interactive editor and terminal interface for programming in the basic language
     SSScmd: edm - context editor for character files
     AMLcmd: qed - programmable context editor for character files

"edm" 1
     SSScmd: edm - context editor for character files

"efficient" 1
     remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines

"element" 2
     SSSsub: ios_$getsize - return current element size for an ioname
     SSSsub: ios_$setsize - set current element size for an ioname

"enable" 1
     AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)

"encoding" 1
     SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname

"endfile" 1
     SSScmd: endfile - close fortran files

"enter" 2
     remark: a user can enter a lower-numbered protection ring only through a "gate" segment
     SSSsub: cu_$cl - call current listener to re-enter command level

"entire" 1
     SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch

"entry" 31
     remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
     remark: a link entry gives the path name of some other entry; it is like an indirect reference
     remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry
     remark: an absolute path name begins with ">" and leads from the root directory to some entry
     SSScmd: addname (an) - add additional entry names to file branch
     SSScmd: deletename (dn) - delete entry name from a file branch
     AMLcmd: fs_chname - add, delete, or change entry names on a file
     SSScmd: link (lk) - create a link entry
     SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
     SSScmd: listnames (ln) - print out entry names for files in a directory
     SSScmd: listotals (lt) - print out entry counts and use for a directory
     SSScmd: rename (rn) - change an entry name for a file
     SSScmd: status (st) - print out status information from directory entry for a branch
     SSScmd: unlink (ul) - delete a link entry from a directory
```

SSSsub: equal_ - expand entry name which may contain equal signs
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and aid user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$append_link - create a link entry in parent directory
SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry
SSSsub: hcs_$set_copysw - set copy switch in branch entry
SSSsub: hcs_$set_relatesw - set relate switch in branch entry
SSSsub: hcs_$star_ - search a directory for entry names matching template
SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
SSSsub: hcs_$status - return information from directory entry about a branch or link
SSSsub: hcs_$status_long - return all information from directory entry about a branch or link

"entry_usage" 2
AMLcmd: entry_usage - print out usage counts for entrypoints
AMLcmd: entry_usage$clear - set entrypoint usage counts to 0

"entrypoint" 7
remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
AMLcmd: entry_usage - print out usage counts for entrypoints
AMLcmd: entry_usage$clear - set entrypoint usage counts to 0
SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)

"eplbsa" 1
AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language

"equal" 1
SSSsub: equal_ - expand entry name which may contain equal signs

"equal_" 1
SSSsub: equal_ - expand entry name which may contain equal signs

"erase" 1
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"error" 8
remark: MPM RefData 1.3 gives all the system error codes
SSScmd: hold - save machine conditions after error
SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
SSSsub: com_err_ - format system error message and write on "user_output"
SSSsub: com_err_$check_fs_errcode_ - return character string for system error code
SSSsub: com_err_$notify - format non-system error message and write on "user_output"
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code

"escape" 1
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"establish" 1
SSSsub: condition_ - establish handler for a condition

```
"event" 21
     remark: an event channel is used to receive inter-process messages and wakeups
     remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
     remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
     SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds
     SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
     SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
     AMLsub: ipc_$block - block process until given event
     AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
     AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
     AMLsub: ipc_$create_ev_chn - create an event channel
     AMLsub: ipc_$cutoff - inhibit event channel for reading
     AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
     AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)
     AMLsub: ipc_$delete_ev_chn - delete an event channel
     AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup
     AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
     AMLsub: ipc_$read_ev_chn - return event message from an event channel
     AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)
     AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
     AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
     AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels

"every" 2
     remark: every known segment in a process is associated with some file or directory branch
     SSScmd: global (gb) - execute a command in every directory inferior to a given directory

"exec_com" 1
     SSScmd: exec_com (ec) - execute a series of command lines contained in a file

"executable" 1
     remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)

"execute" 5
     remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
     SSScmd: basic - compile and execute a program in the basic language
     SSScmd: exec_com (ec) - execute a series of command lines contained in a file
     SSScmd: global (gb) - execute a command in every directory inferior to a given directory
     SSSsub: cu_$cp - call current command processor to execute command line

"execution" 1
     SSScmd: start - restart execution at saved machine conditions

"expand" 4
     SSScmd: get_com_line (gcl) - print out current expanded command line length
     SSScmd: set_com_line (scl) - set maximum expanded command line length
     SSSsub: equal_ - expand entry name which may contain equal signs
     SSSsub: expand_path_ - expand relative path name into absolute path name

"expand_path_" 1
     SSSsub: expand_path_ - expand relative path name into absolute path name

"exponential" 2
     AMLsub: random_$exponential - return random number from exponential distribution with mean 1
     AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution

"exponential_seq" 1
     AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
```

```
"expression" 1
     SSScmd: calc - desk calculator with memory for functions and expressions

"extract" 2
     AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
     AMLcmd: extract_archive - extract all component files from a CTSS archive file

"extract_archive" 1
     AMLcmd: extract_archive - extract all component files from a CTSS archive file
```

Index for Multics - 27 words beginning with f:

"fault" 6
remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signalled
SSScmd: page_trace (pgt) - print out information about last 200 page faults
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_$get_page_trace - return information about last 200 page faults
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval

"file" 58
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand
remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
remark: every known segment in a process is associated with some file or directory branch
SSScmd: addname (an) - add additional entry names to file branch
AMLcmd: adjust - set bit count on file branch
AMLcmd: adjust$block - set bit count on file branch, start from current length
AMLcmd: adjust$block_test - check bit count on file branch, start from current length
AMLcmd: adjust$test - check bit count on file branch
AMLcmd: adjust$test_block - check bit count on file branch, start from current length
SSScmd: archive (ac) - combine several files into an archive file
SSScmd: archive_sort (as) - sort an archive file on component file names
AMLcmd: ascii_check - check all characters of a file for valid ASCII
AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII
AMLcmd: compare_ascii (cpa) - compare two character files
SSScmd: copy (cp) - copy a file
SSScmd: create (cr) - create a file branch
AMLcmd: createSymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
SSScmd: delete (dl) - delete a file
SSScmd: deletedir (dd) - delete a directory branch and all files contained in it
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
SSScmd: deletename (dn) - delete entry name from a file branch
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch
SSScmd: edm - context editor for character files
SSScmd: endfile - close fortran files
SSScmd: exec_com (ec) - execute a series of command lines contained in a file
AMLcmd: extract_archive - extract all component files from a CTSS archive file
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console
AMLcmd: fs_chname - add, delete, or change entry names on a file
SSScmd: help - print out system information files
SSScmd: initiate (in) - make a file known by a given reference name
SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSScmd: listacl (la) - print out ACL for a file
SSScmd: listnames (ln) - print out entry names for files in a directory
AMLcmd: lower_case - convert all characters in a file to lower case
AMLcmd: mail (ml) - send a file to another user's mailbox
SSScmd: moveb (mv) - move a file to another directory
SSScmd: print (pr) - print a character file
AMLcmd: qed - programmable context editor for character files
SSScmd: rename (rn) - change an entry name for a file
SSScmd: reorder_archive - change order of component files in an archive file
AMLcmd: runoff - format a character file for printing
SSScmd: set_bit_count (sbc) - set bit count for a file branch
SSScmd: setacl (sa) - add item to ACL for a file
SSSsub: date_time_$fstime - convert file system time to character string date and time

SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$fs_move_file - move contents from one file branch to another
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$truncate_file - truncate file branch from a given offset
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol

"file_output" 1
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console

"find" 1
remark: "signal_" signals the "unclaimed_signal" condition if it cannot find a handler for a condition

"findata" 1
SSSsub: ti_$findata - turn scratch segment into data segment

"finobj" 1
SSSsub: ti_$finobj - turn scratch segment into object segment

"first" 1
remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called

"fo" 1
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console

"form" 3
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"format" 14
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
remark: MPM RefData 3.5 describes magnetic tape formats
AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
AMLcmd: runoff - format a character file for printing
SSSsub: com_err_ - format system error message and write on "user_output"
SSSsub: com_err_$notify - format non-system error message and write on "user_output"
SSSsub: ioa_ - format a message and write it on "user_output"
SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
SSSsub: ioa_$rs - format a message and return character string
SSSsub: ioa_$rsnnl - format a message and return character string, no newline
AMLsub: read_list_ - read in free-format variables from "user_input"

"fortran" 3
AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
SSScmd: endfile - close fortran files
SSScmd: fortran - compile programs in the fortran language

"frame" 6
AMLcmd: return_to - return to a given stack frame
AMLcmd: return_to$repeat_call - return to given stack frame and repeat call

"GE645" 1
remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645

"GIM" 9
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: hcs_$get_cur_status - return GIM position in DCW list
SSSsub: hcs_$get_status - return GIM status for device channel
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
SSSsub: hcs_$unassign - request GIM to un-assign a device channel

"GIOC" 3
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists

"gate" 3
remark: a user can enter a lower-numbered protection ring only through a "gate" segment
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment

"gb" 1
SSScmd: global (gb) - execute a command in every directory inferior to a given directory

"gcl" 1
SSScmd: get_com_line (gcl) - print out current expanded command line length

"gen_call" 1
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list

"generalized" 1
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system

"get_at_entry_" 1
SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname

"get_com_line" 1
SSScmd: get_com_line (gcl) - print out current expanded command line length

"get_cur_status" 1
SSSsub: hcs_$get_cur_status - return GIM position in DCW list

"get_entry_name" 1
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment

"get_group_id_" 2
SSSsub: get_group_id_ - return user's access control name
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star

"get_page_trace" 1
SSSsub: hcs_$get_page_trace - return information about last 200 page faults

"get_pathname" 1

```
     SSS_af: get_pathname (gpn) - return absolute path name for given reference name

"get_pdir_" 1
     SSSsub: get_pdir_ - return absolute path name of process directory

"get_process_id_" 1
     SSSsub: get_process_id_ - return user process id

"get_ring_" 1
     SSSsub: get_ring_ - return the current ring number

"get_status" 1
     SSSsub: hcs_$get_status - return SIM status for device channel

"getdelim" 1
     SSSsub: ios_$getdelim - return current read delimiters for ioname

"getmode" 1
     SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname

"getquota" 1
     SSScmd: getquota (gq) - print out record quota and current use for given directory

"getseg" 1
     SSSsub: ti_$getseg - set up a scratch segment

"getsize" 1
     SSSsub: ios_$getsize - return current element size for an ioname

"given" 37
     SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console
     SSS_af: get_pathname (gpn) - return absolute path name for given reference name
     SSScmd: getquota (gq) - print out record quota and current use for given directory
     SSScmd: global (gb) - execute a command in every directory inferior to a given directory
     SSScmd: initiate (in) - make a file known by a given reference name
     SSScmd: iocall - call "ios_" with given arguments
     SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
     AMLcmd: return_to - return to a given stack frame
     AMLcmd: return_to$repeat_call - return to given stack frame and repeat call
     SSScmd: terminate_refname (tmr) - terminate a segment given by reference name
     SSScmd: terminate_segno (tms) - terminate a segment given by segment number
     SSSsub: acm_$sleep_time - block process until time given in microseconds
     SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
     SSSsub: alloc_ - allocate storage in given area, return pointer
     SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
     SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments
     SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
     SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
     SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
     SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
     SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
     SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
     SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds
     SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
     SSSsub: hcs_$truncate_file - truncate file branch from a given offset
     SSSsub: hcs_$truncate_seg - truncate segment from a given offset
     AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id
     SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
```

AMLsub: ipc_$block - block process until given event
SSSsub: move_ - move a block of words from one place to another (given by pointers)
SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
AMLsub: reverse_index - scan character string from right to left for given character
SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
SSSsub: term_$refname - same as term_ for segment given by reference name
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer
AMLsub: whoami_ - return user's name and project as given at login

"gives" 4
remark: a link entry gives the path name of some other entry; it is like an indirect reference
remark: each item on an ACL gives an access name and the access mode permitted
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences

"giving" 2
remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory

"global" 1
SSScmd: global (gb) - execute a command in every directory inferior to a given directory

"gpn" 1
SSS_af: get_pathname (gpn) - return absolute path name for given reference name

"gq" 1
SSScmd: getquota (gq) - print out record quota and current use for given directory

"group" 3
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number

"grouped" 1
remark: the segments addressable by a process are grouped into rings of protection

"grow_stack_frame" 1
SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame

Index for Multics - 10 words beginning with h

"handler" 4
remark: "signal_" signals the "unclaimed_signal" condition if it cannot find a handler for a condition
SSSsub: condition_ - establish handler for a condition
SSSsub: reversion_ - revert a condition handler
SSSsub: signal_ - signal the occurrence of a condition, call its handler

"hardcore" 1
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor

"hcs_" 63
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$append_link - create a link entry in parent directory
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: hcs_$assign_linkage - allocate storage in linkage section
AMLsub: hcs_$block - block process until next wakeup
SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_$fs_get_mode - return access of user with respect to segment
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_$fs_move_file - move contents from one file branch to another
SSSsub: hcs_$fs_move_seg - move contents from one segment to another
SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_$fs_search_set_wdir - change current working directory
SSSsub: hcs_$get_cur_status - return GIM position in DCW list
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_$get_page_trace - return information about last 200 page faults
SSSsub: hcs_$get_status - return GIM status for device channel
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry

```
     SSSsub: hcs_$set_relatesw - set relate switch in branch entry
     SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
     SSSsub: hcs_$star_ - search a directory for entry names matching template
     SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
     SSSsub: hcs_$status - return information from directory entry about a branch or link
     SSSsub: hcs_$status_long - return all information from directory entry about a branch or link
     SSSsub: hcs_$status_minf - return bit count and type for a branch or link
     SSSsub: hcs_$status_mins - return bit count and type for a segment
     SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
     SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
     SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
     SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown
     SSSsub: hcs_$truncate_file - truncate file branch from a given offset
     SSSsub: hcs_$truncate_seg - truncate segment from a given offset
     SSSsub: hcs_$unassign - request GIM to unassign a device channel
     SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used
     AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id

"help" 1
     SSScmd: help - print out system information files

"high" 2
     SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
     SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch

"hmu" 1
     SSScmd: how_many_users (hmu) - print out current number of users and maximum number

"hold" 2
     SSScmd: hold - save machine conditions after error
     SSScmd: release - release machine conditions saved by "hold"

"home" 3
     remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
     SSScmd: printhomedir (phd) - print absolute path name of home directory
     SSScmd: sethomedir (shd) - set path name of home directory

"horizontal" 1
     AMLcmd: pht - print horizontal tab

"how_many_users" 1
     SSScmd: how_many_users (hmu) - print out current number of users and maximum number
```

```
"INTERRUPT" 1
     remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)

"id" 5
     SSScmd: who$admin - print out names, project id's, and process id's for users on system
     SSScmd: who$long - print out names and project id's for users on system
     SSSsub: get_process_id_ - return user process id
     SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
     AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id

"identify" 1
     SSScmd: login - identify user by name and project to system control and create user process

"increase" 1
     SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame

"ind" 1
     AMLcmd: ind - indent programs in the pl1 language

"indent" 1
     AMLcmd: ind - indent programs in the pl1 language

"index" 5
     remark: SSS commands are marked "SSScmd" in this index; consult the SSS section of the MPM
     remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
     remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
     remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
     remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM

"indices" 2
     SSSsub: parse_file_ - scan character file and return indices of next break or symbol
     SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file

"indirect" 1
     remark: a link entry gives the path name of some other entry; it is like an indirect reference

"inferior" 4
     remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
     SSScmd: global (gb) - execute a command in every directory inferior to a given directory
     SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
     SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory

"information" 9
     remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645
     SSScmd: help - print out system information files
     SSScmd: page_trace (pgt) - print out information about last 200 page faults
     SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
     SSScmd: status (st) - print out status information from directory entry for a branch
     SSSsub: hcs_$get_page_trace - return information about last 200 page faults
     SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
     SSSsub: hcs_$status - return information from directory entry about a branch or link
     SSSsub: hcs_$status_long - return all information from directory entry about a branch or link

"inhibit" 1
     AMLsub: ipc_$cutoff - inhibit event channel for reading
```

"initialize" 4
SSSsub: area_ - initialize an area
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment

"initiate" 2
SSScmd: initiate (in) - make a file known by a given reference name
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer

"initiate_count" 1
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count

"input" 5
remark: an "ioname" is the name of an input-output stream or device channel
remark: users make calls to the input-output switch "ios_" to read and write on ionames
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 1.6 describes punch-card input and control cards

"instance" 2
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star

"instead" 1
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console

"instruction" 1
remark: a "process" is a locus of control within an instruction sequence

"inter" 4
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
remark: an event channel is used to receive inter-process messages and wakeups
remark: MPM RefData 3.4 describes the linkage section and inter-segment references
AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id

"interactive" 3
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language
AMLcmd: debug (db) - symbolic interactive debugging aid
SSScmd: probe - interactive debugging aid

"interface" 2
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language

"interrogate" 4
AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels

"interval" 5
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds
SSSsub: acm_$sleep_interval - block process for time interval in microseconds
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds
SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval

"ioa_" 6

```
     SSSsub: ioa_ - format a message and write it on "user_output"
     SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
     SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
     SSSsub: ioa_$rs - format a message and return character string
     SSSsub: ioa_$rsnnl - format a message and return character string, no newline

"ioa_stream" 1
     SSSsub: ioa_$ioa_stream - format a message and write it on given ioname

"ioa_stream_nnl" 1
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline

"iocall" 1
     SSScmd: iocall - call "ios_" with given arguments

"iomode" 1
     SSScmd: iomode - set character conversion mode for an ioname

"ioname" 32
     remark: an "ioname" is the name of an input-output stream or device channel
     remark: users make calls to the input-output switch "ios_" to read and write on ionames
     SSScmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"
     SSScmd: iomode - set character conversion mode for an ioname
     SSScmd: line_length (ll) - set maximum line length for an ioname
     SSScmd: print_attach_table (pat) - print ioname attachments for process
     SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
     SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
     SSSsub: ios_$abort - abort outstanding transactions for an ioname
     SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
     SSSsub: ios_$changemode - change mode of attachment of an ioname
     SSSsub: ios_$detach - un-attach one ioname from another ioname
     SSSsub: ios_$getdelim - return current read delimiters for ioname
     SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname
     SSSsub: ios_$getsize - return current element size for an ioname
     SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
     SSSsub: ios_$order - issue request to outer module associated with an ioname
     SSSsub: ios_$read - read from an ioname
     SSSsub: ios_$read_ptr - read from ioname "user_input"
     SSSsub: ios_$readsync - set read synchronization mode for an ioname
     SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
     SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname
     SSSsub: ios_$seek - set reference pointers associated with an ioname
     SSSsub: ios_$setdelim - set current read delimiters for an ioname
     SSSsub: ios_$setsize - set current element size for an ioname
     SSSsub: ios_$tell - return value of reference pointer associated with an ioname
     SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname
     SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
     SSSsub: ios_$write - write on an ioname
     SSSsub: ios_$write_ptr - write on ioname "user_output"
     SSSsub: ios_$writesync - set write synchronization mode for an ioname

"ios_" 25
     remark: users make calls to the input-output switch "ios_" to read and write on ionames
     SSScmd: iocall - call "ios_" with given arguments
     SSSsub: ios_$abort - abort outstanding transactions for an ioname
     SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
     SSSsub: ios_$changemode - change mode of attachment of an ioname
```

SSSsub: ios_$detach - un-attach one ioname from another ioname
SSSsub: ios_$getdelim - return current read delimiters for ioname
SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname
SSSsub: ios_$getsize - return current element size for an ioname
SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
SSSsub: ios_$order - issue request to outer module associated with an ioname
SSSsub: ios_$read - read from an ioname
SSSsub: ios_$read_ptr - read from ioname "user_input"
SSSsub: ios_$readsync - set read synchronization mode for an ioname
SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname
SSSsub: ios_$seek - set reference pointers associated with an ioname
SSSsub: ios_$setdelim - set current read delimiters for an ioname
SSSsub: ios_$setsize - set current element size for an ioname
SSSsub: ios_$tell - return value of reference pointer associated with an ioname
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname
SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
SSSsub: ios_$write - write on an ioname
SSSsub: ios_$write_ptr - write on ioname "user_output"
SSSsub: ios_$writesync - set write synchronization mode for an ioname

"iowait" 1
SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname

"ipc_" 16
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
AMLsub: ipc_$block - block process until given event
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
AMLsub: ipc_$create_ev_chn - create an event channel
AMLsub: ipc_$cutoff - inhibit event channel for reading
AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)
AMLsub: ipc_$delete_ev_chn - delete an event channel
AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup
AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
AMLsub: ipc_$read_ev_chn - return event message from an event channel
AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels

"issue" 1
SSSsub: ios_$order - issue request to outer module associated with an ioname

"item" 9
remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch
remark: each item on an ACL gives an access name and the access mode permitted
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
SSScmd: deleteacl (da) - delete item from an ACL
SSScmd: deletecacl (dc) - delete item from a CACL
SSScmd: setacl (sa) - add item to ACL for a file
SSScmd: setcacl (sc) - add item to CACL for a directory
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch

Index for Multics - 4 words beginning with k

"KST" 10
remark: the KST (known segment table) is a per-process table of reference names and segment numbers
SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown

"kept" 1
remark: each process has a process directory, where per-process segments are kept

"kill" 1
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"known" 9
remark: every known segment in a process is associated with some file or directory branch
remark: the KST (known segment table) is a per-process table of reference names and segment numbers
SSScmd: initiate (in) - make a file known by a given reference name
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown

Index for Multics - 42 words beginning with l.

"Library" 1
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS

"la" 1
SSScmd: listacl (la) - print out ACL for a file

"language" 10
SSScmd: basic - compile and execute a program in the basic language
AMLcmd: bcpl - compile a program in the bcpl language
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language
AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
AMLcmd: createSymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language
SSScmd: fortran - compile programs in the fortran language
AMLcmd: ind - indent programs in the pl1 language
SSScmd: pl1 - compile programs in the pl1 language
SSScmd: pl1d: request delayed compile of programs in the pl1 language

"last" 3
SSScmd: page_trace (pgt) - print out information about last 200 page faults
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_$get_page_trace - return information about last 200 page faults

"lc" 1
SSScmd: listcacl (lc) - print out CACL for a directory

"leads" 1
remark: an absolute path name begins with ">" and leads from the root directory to some entry

"left" 1
AMLsub: reverse_index - scan character string from right to left for given character

"length" 8
AMLcmd: adjust$block - set bit count on file branch, start from current length
AMLcmd: adjust$block_test - check bit count on file branch, start from current length
AMLcmd: adjust$test_block - check bit count on file branch, start from current length
SSScmd: get_com_line (gcl) - print out current expanded command line length
SSScmd: line_length (ll) - set maximum line length for an ioname
SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSScmd: set_com_line (scl) - set maximum expanded command line length
SSSsub: area_$redef - change the length of an area

"ler" 1
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system

"level" 5
SSSsub: cu_$cl - call current listener to re-enter command level
SSSsub: cu_$level_get - return current validation level
SSSsub: cu_$level_set - set validation level
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2

"level_get" 1
SSSsub: cu_$level_get - return current validation level

"level_set" 1

```
     SSSsub: cu_$level_set - set validation level

"like" 1
     remark: a link entry gives the path name of some other entry; it is like an indirect reference

"limit" 1
     remark: a directory may have a record quota, which limits the number of pages which may be charged to it

"line" 6
     SSScmd: exec_com (ec) - execute a series of command lines contained in a file
     SSScmd: get_com_line (gcl) - print out current expanded command line length
     SSScmd: line_length (ll) - set maximum line length for an ioname
     SSScmd: set_com_line (scl) - set maximum expanded command line length
     SSSsub: cu_$cp - call current command processor to execute command line
     SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file

"line_length" 1
     SSScmd: line_length (ll) - set maximum line length for an ioname

"link" 11
     remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
     remark: a link entry gives the path name of some other entry; it is like an indirect reference
     SSScmd: link (lk) - create a link entry
     SSScmd: unlink (ul) - delete a link entry from a directory
     SSSsub: hcs_$append_link - create a link entry in parent directory
     SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
     SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
     SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
     SSSsub: hcs_$status - return information from directory entry about a branch or link
     SSSsub: hcs_$status_long - return all information from directory entry about a branch or link
     SSSsub: hcs_$status_minf - return bit count and type for a branch or link

"linkage" 10
     remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
     remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
     remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signalled
     remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
     remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
     remark: MPM RefData 3.4 describes the linkage section and inter-segment references
     SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
     SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
     SSSsub: hcs_$assign_linkage - allocate storage in linkage section
     SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment

"linkage_error" 1
     remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signalled

"linker" 3
     remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
     remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
     remark: MPM RefData 2.7 describes the system standard search rules used by the linker

"list" 13
     remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch
     remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
     remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
     remark: MPM RefData 2.2.1 lists all condition names which the system may signal
     remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
     SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
```

SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
SSSsub: hcs_$get_cur_status - return GIM position in DCW list
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list

"list_change" 1
SSSsub: hcs_$list_change - request GIM to change DCW list

"list_connect" 1
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list

"list_size" 1
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel

"listacl" 1
SSScmd: listacl (la) - print out ACL for a file

"listcacl" 1
SSScmd: listcacl (lc) - print out CACL for a directory

"listener" 5
SSScmd: ready (rdy) - print listener ready message
SSScmd: ready_off (rif) - turn off listener ready messages
SSScmd: ready_on (rin) - turn on listener ready messages
SSSsub: cu_$cl - call current listener to re-enter command level
SSSsub: cu_$set_cl - define current listener

"listnames" 1
SSScmd: listnames (ln) - print out entry names for files in a directory

"listotals" 1
SSScmd: listotals (lt) - print out entry counts and use for a directory

"lk" 1
SSScmd: link (lk) - create a link entry

"ll" 1
SSScmd: line_length (ll) - set maximum line length for an ioname

"ln" 1
SSScmd: listnames (ln) - print out entry names for files in a directory

"local" 1
remark: MPM RefData 2.1 describes local conventions for Multics users

"lock" 1
AMLcmd: mail$unlock - unlock a locked mailbox

"locus" 1
remark: a "process" is a locus of control within an instruction sequence

"login" 4
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
remark: MPM RefData 1.1 describes the login and logout commands and system messages
SSScmd: login - identify user by name and project to system control and create user process
AMLsub: whoami_ - return user's name and project as given at login

```
"logout" 2
     remark: MPM RefData 1.1 describes the login and logout commands and system messages
     SSScmd: logout - request process destruction and session termination

"long" 2
     SSScmd: who$long - print out names and project id's for users on system
     SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information

"lower" 2
     remark: a user can enter a lower-numbered protection ring only through a "gate" segment
     AMLcmd: lower_case - convert all characters in a file to lower case

"lower_case" 1
     AMLcmd: lower_case - convert all characters in a file to lower case

"ls" 1
     SSScmd: list (ls) - print out access, length, and entry names for given files in a directory

"lt" 1
     SSScmd: listotals (lt) - print out entry counts and use for a directory
```

Index for Multics - 37 words beginning with m

"MPM" 21
remark: the Multics Programmer's Manual (MPM) is the reference manual for the system
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 1.1 describes the login and logout commands and system messages
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 1.6 describes punch-card input and control cards
remark: MPM RefData 2.1 describes local conventions for Multics users
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
remark: MPM RefData 3.4 describes the linkage section and inter-segment references
remark: MPM RefData 3.5 describes magnetic tape formats
remark: MPM RefData 3.7 describes punched card codes
remark: SSS commands are marked "SSScmd" in this index; consult the SSS section of the MPM
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM

"Maintained" 1
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS

"Manual" 1
remark: the Multics Programmer's Manual (MPM) is the reference manual for the system

"Multics" 5
remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645
remark: the Multics Programmer's Manual (MPM) is the reference manual for the system
remark: a user normally controls his Multics process from a typewriter terminal (console)
remark: MPM RefData 2.1 describes local conventions for Multics users
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format

"machine" 4
SSScmd: hold - save machine conditions after error
SSScmd: release - release machine conditions saved by "hold"
SSScmd: start - restart execution at saved machine conditions
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0

"magnetic" 1
remark: MPM RefData 3.5 describes magnetic tape formats

"mail" 2
AMLcmd: mail (ml) - send a file to another user's mailbox
AMLcmd: mail$unlock - unlock a locked mailbox

"mailbox" 2
AMLcmd: mail (ml) - send a file to another user's mailbox
AMLcmd: mail$unlock - unlock a locked mailbox

"maintained" 1
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link

```
     remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
     remark: MPM RefData 1.1 describes the login and logout commands and system messages
     SSScmd: ready (rdy) - print listener ready message
     SSScmd: ready_off (rif) - turn off listener ready messages
     SSScmd: ready_on (rin) - turn on listener ready messages
     SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
     SSSsub: com_err_ - format system error message and write on "user_output"
     SSSsub: com_err_$notify - format non-system error message and write on "user_output"
     SSSsub: ioa_ - format a message and write it on "user_output"
     SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
     SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
     SSSsub: ioa_$rs - format a message and return character string
     SSSsub: ioa_$rsnnl - format a message and return character string, no newline
     AMLsub: ipc_$read_ev_chn - return event message from an event channel

"microsecond" 6
     SSSsub: acm_$sleep_interval - block process for time interval in microseconds
     SSSsub: acm_$sleep_time - block process until time given in microseconds
     SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
     SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
     SSSsub: clock_ - return calendar clock time in microseconds
     SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds

"ml" 1
     AMLcmd: mail (ml) - send a file to another user's mailbox

"mode" 10
     remark: each item on an ACL gives an access name and the access mode permitted
     remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
     SSScmd: iomode - set character conversion mode for an ioname
     SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
     SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
     SSSsub: ios_$changemode - change mode of attachment of an ioname
     SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname
     SSSsub: ios_$readsync - set read synchronization mode for an ioname
     SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
     SSSsub: ios_$writesync - set write synchronization mode for an ioname

"model" 1
     remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)

"module" 5
     remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
     remark: MPM RefData 1.4 describes the input-output system and device and outer module types
     SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
     SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
     SSSsub: ios_$order - issue request to outer module associated with an ioname

"move" 5
     SSScmd: moveb (mv) - move a file to another directory
     SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
     SSSsub: hcs_$fs_move_file - move contents from one file branch to another
     SSSsub: hcs_$fs_move_seg - move contents from one segment to another
     SSSsub: move_ - move a block of words from one place to another (given by pointers)

"move_" 1
     SSSsub: move_ - move a block of words from one place to another (given by pointers)
```

"moveb" 1
SSScmd: moveb (mv) - move a file to another directory

"movequota" 1
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory

"mq" 1
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory

"multiple" 4
AMLsub: roundb - round a number to next multiple of 2**n
AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
AMLsub: roundb$rounib36 - round a number to multiple of 36, round to multiple of 2**n

"multiplex" 1
remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645

"must" 1
remark: a "command" is a procedure which can be called from the console; its arguments must be character strings

"mv" 1
SSScmd: moveb (mv) - move a file to another directory

"name" 63
remark: a "segment" is an addressable collection of words which has names and access attributes
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
remark: a link entry gives the path name of some other entry; it is like an indirect reference
remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry
remark: an absolute path name begins with ">" and leads from the root directory to some entry
remark: a relative path name is relative to the current working directory; it begins with "<" or a name
remark: the KST (known segment table) is a per-process table of reference names and segment numbers
remark: each item on an ACL gives an access name and the access mode permitted
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
remark: an "ioname" is the name of an input-output stream or device channel
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
SSScmd: addname (an) - add additional entry names to file branch
SSScmd: archive_sort (as) - sort an archive file on component file names
SSScmd: deletename (dn) - delete entry name from a file branch
AMLcmd: fs_chname - add, delete, or change entry names on a file
SSS_af: get_pathname (gpn) - return absolute path name for given reference name
SSScmd: initiate (in) - make a file known by a given reference name
SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSScmd: listnames (ln) - print out entry names for files in a directory
SSScmd: login - identify user by name and project to system control and create user process
SSS_af: pd - return absolute path name of process directory
SSScmd: printhomedir (phd) - print absolute path name of home directory
SSScmd: printwdir (pwd) - print absolute path name of current working directory
SSScmd: rename (rn) - change an entry name for a file
SSScmd: sethomedir (shd) - set path name of home directory
SSScmd: terminate (tm) - terminate a reference name
SSScmd: terminate_refname (tmr) - terminate a segment given by reference name
SSS_af: wd - return absolute path name of current working directory
SSScmd: who - print out names of users on system
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSScmd: who$long - print out names and project id's for users on system
SSSsub: check_star_ - check whether path name contains a star
SSSsub: equal_ - expand entry name which may contain equal signs
SSSsub: expand_path_ - expand relative path name into absolute path name
SSSsub: get_group_id_ - return user's access control name
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star
SSSsub: get_pdir_ - return absolute path name of process directory
SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
SSSsub: hcs_$star_ - search a directory for entry names matching template
SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown

```
     SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
     SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
     SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown
     AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
     AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
     SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
     SSSsub: term_$refname - same as term_ for segment given by reference name
     AMLsub: whoami_ - return user's name and project as given at login

"necessary" 1
     SSScmd: deleteforce (df) - delete a file branch; set write access if necessary

"new" 1
     SSScmd: new_proc - request user control to create a new user process

"new_proc" 1
     SSScmd: new_proc - request user control to create a new user process

"newline" 4
     AMLcmd: pnl - print newline
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
     SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
     SSSsub: ioa_$rsnnl - format a message and return character string, no newline

"next" 4
     AMLsub: hcs_$block - block process until next wakeup
     SSSsub: parse_file_ - scan character file and return indices of next break or symbol
     SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
     AMLsub: roundb - round a number to next multiple of 2**n

"nnl" 1
     SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline

"no" 3
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
     SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
     SSSsub: ioa_$rsnnl - format a message and return character string, no newline

"nomakeunknown" 1
     SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate

"normal" 4
     AMLsub: random_$normal - return random number from normal distribution
     AMLsub: random_$normal_ant - return random number from antithetic normal distribution
     AMLsub: random_$normal_seq - return array of random numbers from normal distribution
     AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution

"normal_ant" 1
     AMLsub: random_$normal_ant - return random number from antithetic normal distribution

"normal_ant_seq" 1
     AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution

"normal_seq" 1
     AMLsub: random_$normal_seq - return array of random numbers from normal distribution

"normally" 1
     remark: a user normally controls his Multics process from a typewriter terminal (console)
```

"notify" 1
SSSsub: com_err_$notify - format non-system error message and write on "user_output"

"null" 2
remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST

"number" 24
remark: the KST (known segment table) is a per-process table of reference names and segment numbers
remark: a user can enter a lower-numbered protection ring only through a "gate" segment
remark: a directory may have a record quota, which limits the number of pages which may be charged to it
SSScmd: how_many_users (hmu) - print out current number of users and maximum number
SSScmd: terminate_segno (tms) - terminate a segment given by segment number
SSSsub: cu_$arg_count - return number of arguments of calling procedure
SSSsub: get_ring_ - return the current ring number
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
AMLsub: random_$exponential - return random number from exponential distribution with mean 1
AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
AMLsub: random_$normal - return random number from normal distribution
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_seq - return array of random numbers from normal distribution
AMLsub: random_$random_get_seed - return current seed for random numbers
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$random_set_seed - set seed for random numbers
AMLsub: random_$uniform - return random number from uniform distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution
AMLsub: roundb - round a number to next multiple of 2**n
AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
AMLsub: roundb$roundb36 - round a number to multiple of 36, round to multiple of 2**n

"object" 7
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments
SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
SSScmd: print_object_map (pom) - print object map for an object segment
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
SSSsub: make_obj_map_ - write an object map into object segment
SSSsub: ti_$finobj - turn scratch segment into object segment

"occurrence" 1
SSSsub: signal_ - signal the occurrence of a condition, call its handler

"occurs" 1
remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called

"oct" 1
SSSsub: cv_bin_$oct - convert binary to character string, base 8

"off" 1
SSScmd: ready_off (rdf) - turn off listener ready messages

"offset" 3
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_$truncate_file - truncate file branch from a given offset
SSSsub: hcs_$truncate_seg - truncate segment from a given offset

"only" 1
remark: a user can enter a lower-numbered protection ring only through a "gate" segment

"order" 2
SSScmd: reorder_archive - change order of component files in an archive file
SSSsub: ios_$order - issue request to outer module associated with an ioname

"other" 1
remark: a link entry gives the path name of some other entry; it is like an indirect reference

"ourter" 1
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types

"outer" 3
SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSsub: ios_$order - issue request to outer module associated with an ioname

"output" 4
remark: an "ioname" is the name of an input-output stream or device channel
remark: users make calls to the input-output switch "ios_" to read and write on inames
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types

"outstanding" 1
SSSsub: ios_$abort - abort outstanding transactions for an ioname

"Programmer" 1
remark: the Multics Programmer's Manual (MPM) is the reference manual for the system

"page" 6
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand
remark: a directory may have a record quota, which limits the number of pages which may be charged to it
remark: if the record quota of a directory is 0, its pages are charged to its parent directory
SSScmd: page_trace (pgt) - print out information about last 200 page faults
SSSsub: hcs_$get_page_trace - return information about last 200 page faults
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used

"page_trace" 1
SSScmd: page_trace (pgt) - print out information about last 200 page faults

"parent" 9
remark: if the record quota of a directory is 0, its pages are charged to its parent directory
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$append_link - create a link entry in parent directory
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment

"parse_file_" 7
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: parse_file_$parse_file_set_break - declare break characters to parse_file_
SSSsub: parse_file_$parse_file_unset_break - undeclare break characters for parse_file_

"parse_file_cur_line" 1
SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file

"parse_file_init" 1
SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name

"parse_file_init_ptr" 1
SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment

"parse_file_ptr" 1
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol

"parse_file_set_break" 1
SSSsub: parse_file_$parse_file_set_break - declare break characters to parse_file_

"parse_file_unset_break" 1
SSSsub: parse_file_$parse_file_unset_break - undeclare break characters for parse_file_

"part" 3
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch

```
"pat" 1
     SSScmd: print_attach_table (pat) - print ioname attachments for process

"path" 17
     remark: a link entry gives the path name of some other entry; it is like an indirect reference
     remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry
     remark: an absolute path name begins with ">" and leads from the root directory to some entry
     remark: a relative path name is relative to the current working directory; it begins with "<" or a name
     SSS_af: get_pathname (gpn) - return absolute path name for given reference name
     SSS_af: pd - return absolute path name of process directory
     SSScmd: printhomedir (phd) - print absolute path name of home directory
     SSScmd: printwdir (pwd) - print absolute path name of current working directory
     SSScmd: sethomedir (shd) - set path name of home directory
     SSS_af: wd - return absolute path name of current working directory
     SSSsub: check_star_ - check whether path name contains a star
     SSSsub: expand_path_ - expand relative path name into absolute path name
     SSSsub: get_pdir_ - return absolute path name of process directory
     SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
     SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
     SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
     SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST

"pd" 1
     SSS_af: pd - return absolute path name of process directory

"pending" 1
     AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup

"per" 3
     remark: each process has a process directory, where per-process segments are kept
     remark: the KST (known segment table) is a per-process table of reference names and segment numbers
     remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch

"perform" 1
     remark: a DCW (data control word) requests the GIOC to perform some action on a device channel

"peripheral" 1
     remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system

"permit" 1
     remark: each item on an ACL gives an access name and the access mode permitted

"pgt" 1
     SSScmd: page_trace (pgt) - print out information about last 200 page faults

"phd" 1
     SSScmd: printhomedir (phd) - print absolute path name of home directory

"pht" 1
     AMLcmd: pht - print horizontal tab

"pi" 1
     SSScmd: program_interrupt (pi) - signal the condition "program_interrupt"

"pl1" 5
     AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
     AMLcmd: ind - indent programs in the pl1 language
     SSScmd: pl1 - compile programs in the pl1 language
```

SSScmd: pl1d: request delayed compile of programs in the pl1 language
AMLcmd: printsymtab (pst) - print symbol table for a pl1 program

"pl1d" 1
SSScmd: pl1d: request delayed compile of programs in the pl1 language

"place" 1
SSSsub: move_ - move a block of words from one place to another (given by pointers)

"pli" 1
SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment

"pnl" 1
AMLcmd: pnl - print newline

"pointer" 20
remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
SSSsub: alloc_ - allocate storage in given area, return pointer
SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
SSSsub: cu_$arg_ptr - return pointer to character-string argument
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments
SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: ios_$seek - set reference pointers associated with an ioname
SSSsub: ios_$tell - return value of reference pointer associated with an ioname
SSSsub: move_ - move a block of words from one place to another (given by pointers)
SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer

"pom" 1
SSScmd: print_object_map (pom) - print object map for an object segment

"position" 1
SSSsub: hcs_$get_cur_status - return IPM position in DCW list

"pr" 1
SSScmd: print (pr) - print a character file

"print" 28
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
AMLcmd: entry_usage - print out usage counts for entrypoints
SSScmd: get_com_line (gcl) - print out current expanded command line length
SSScmd: getquota (gq) - print out record quota and current use for given directory
SSScmd: help - print out system information files
SSScmd: how_many_users (hmu) - print out current number of users and maximum number
SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSScmd: listacl (la) - print out ACL for a file
SSScmd: listcacl (lc) - print out CACL for a directory
SSScmd: listnames (ln) - print out entry names for files in a directory
SSScmd: listotals (lt) - print out entry counts and use for a directory
SSScmd: page_trace (pgt) - print out information about last 200 page faults

```
     AMLcmd: pht - print horizontal tab
     AMLcmd: pnl - print newline
     SSScmd: print (pr) - print a character file
     SSScmd: print_attach_table (pat) - print ioname attachments for process
     SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
     SSScmd: print_object_map (pom) - print object map for an object segment
     SSScmd: printhomedir (phd) - print absolute path name of home directory
     AMLcmd: printsymtab (pst) - print symbol table for a pl1 program
     SSScmd: printwdir (pwd) - print absolute path name of current working directory
     AMLcmd: ps - print argument string
     SSScmd: ready (rdy) - print listener ready message
     AMLcmd: runoff - format a character file for printing
     SSScmd: status (st) - print out status information from directory entry for a branch
     SSScmd: who - print out names of users on system
     SSScmd: who$admin - print out names, project id's, and process id's for users on system
     SSScmd: who$long - print out names and project id's for users on system

"print_attach_table" 1
     SSScmd: print_attach_table (pat) - print ioname attachments for process

"print_link_info" 1
     SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment

"print_object_map" 1
     SSScmd: print_object_map (pom) - print object map for an object segment

"printer" 1
     SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer

"printhomedir" 1
     SSScmd: printhomedir (phd) - print absolute path name of home directory

"printsymtab" 1
     AMLcmd: printsymtab (pst) - print symbol table for a pl1 program

"printwdir" 1
     SSScmd: printwdir (pwd) - print absolute path name of current working directory

"probe" 1
     SSScmd: probe - interactive debugging aid

"proc_info" 1
     SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number

"procedure" 11
     remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
     remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
     remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
     SSSsub: cu_$arg_count - return number of arguments of calling procedure
     SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
     SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
     SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame
     SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments
     SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
     SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
     SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame

"process" 25
     remark: a user normally controls his Multics process from a typewriter terminal (console)
```

remark: a "process" is a locus of control within an instruction sequence
remark: a process has a two-dimensional address space consisting of segments
remark: every known segment in a process is associated with some file or directory branch
remark: each process has a process directory, where per-process segments are kept
remark: the KST (known segment table) is a per-process table of reference names and segment numbers
remark: the segments addressable by a process are grouped into rings of protection
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
remark: an event channel is used to receive inter-process messages and wakeups
SSScmd: login - identify user by name and project to system control and create user process
SSScmd: logout - request process destruction and session termination
SSScmd: new_proc - request user control to create a new user process
SSS_af: pd - return absolute path name of process directory
SSScmd: print_attach_table (pat) - print ioname attachments for process
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds
SSSsub: acm_$sleep_interval - block process for time interval in microseconds
SSSsub: acm_$sleep_time - block process until time given in microseconds
SSSsub: get_pdir_ - return absolute path name of process directory
SSSsub: get_process_id_ - return user process id
AMLsub: hcs_$block - block process until next wakeup
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id
AMLsub: ipc_$block - block process until given event

"processor" 2
SSSsub: cu_$cp - call current command processor to execute command line
SSSsub: cu_$set_cp - define current command processor

"program" 14
remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
SSScmd: basic - compile and execute a program in the basic language
AMLcmd: bcpl - compile a program in the bcpl language
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language
AMLcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language
SSScmd: fortran - compile programs in the fortran language
AMLcmd: ind - indent programs in the pl1 language
SSScmd: pl1 - compile programs in the pl1 language
SSScmd: pl1d: request delayed compile of programs in the pl1 language
AMLcmd: printsymtab (pst) - print symbol table for a pl1 program
AMLcmd: qed - programmable context editor for character files

"program_interrupt" 1
SSScmd: program_interrupt (pi) - signal the condition "program_interrupt"

"project" 6
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
SSScmd: login - identify user by name and project to system control and create user process
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSScmd: who$long - print out names and project id's for users on system
AMLsub: whoami_ - return user's name and project as given at login

"protection" 2
remark: the segments addressable by a process are grouped into rings of protection
remark: a user can enter a lower-numbered protection ring only through a "gate" segment

```
"ps" 1
     AMLcmd: ps - print argument string

"pst" 1
     AMLcmd: printsymtab (pst) - print symbol table for a pl1 program

"ptr_call" 1
     SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments

"punch" 3
     remark: MPM RefData 1.6 describes punch-card input and control cards
     remark: MPM RefData 3.7 describes punched card codes
     SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch

"pwd" 1
     SSScmd: printwdir (pwd) - print absolute path name of current working directory
```

Index for Multics = 5 words beginning with q

"qed" 1
AMLcmd: qed = programmable context editor for character files

"qualify" 1
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS

"quit" 1
remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)

"quota" 5
remark: a directory may have a record quota, which limits the number of pages which may be charged to it
remark: if the record quota of a directory is 0, its pages are charged to its parent directory
SSScmd: getquota (gq) = print out record quota and current use for given directory
SSScmd: movequota (mq) = move record quota between parent directory and inferior directory
SSSsub: hcs_$quota_get = return record quota, current use, and cumulative use for directory branch

"quota_get" 1
SSSsub: hcs_$quota_get = return record quota, current use, and cumulative use for directory branch

"RefData" 15
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 1.1 describes the login and logout commands and system messages
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 1.6 describes punch-card input and control cards
remark: MPM RefData 2.1 describes local conventions for Multics users
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
remark: MPM RefData 3.4 describes the linkage section and inter-segment references
remark: MPM RefData 3.5 describes magnetic tape formats
remark: MPM RefData 3.7 describes punched card codes

"random" 12
AMLsub: random_$exponential - return random number from exponential distribution with mean 1
AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
AMLsub: random_$normal - return random number from normal distribution
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_seq - return array of random numbers from normal distribution
AMLsub: random_$random_get_seed - return current seed for random numbers
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$random_set_seed - set seed for random numbers
AMLsub: random_$uniform - return random number from uniform distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution

"random_" 12
AMLsub: random_$exponential - return random number from exponential distribution with mean 1
AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
AMLsub: random_$normal - return random number from normal distribution
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_seq - return array of random numbers from normal distribution
AMLsub: random_$random_get_seed - return current seed for random numbers
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$random_set_seed - set seed for random numbers
AMLsub: random_$uniform - return random number from uniform distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution

"random_get_seed" 1
AMLsub: random_$random_get_seed - return current seed for random numbers

"random_set_seed" 1
AMLsub: random_$random_set_seed - set seed for random numbers

"rdf" 1
SSScmd: ready_off (rdf) - turn off listener ready messages

"rdn" 1
SSScmd: ready_on (rdn) - turn on listener ready messages

```
"rdy" 1
     SSScmd: ready (rdy) - print listener ready message

"read" 9
     remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
     remark: users make calls to the input-output switch "ios_" to read and write on ionames
     SSSsub: ios_$getdelim - return current read delimiters for ioname
     SSSsub: ios_$read - read from an ioname
     SSSsub: ios_$read_ptr - read from ioname "user_input"
     SSSsub: ios_$readsync - set read synchronization mode for an ioname
     SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
     SSSsub: ios_$setdelim - set current read delimiters for an ioname
     AMLsub: read_list_ - read in free-format variables from "user_input"

"read_ev_chn" 1
     AMLsub: ipc_$read_ev_chn - return event message from an event channel

"read_list_" 1
     AMLsub: read_list_ - read in free-format variables from "user_input"

"read_ptr" 1
     SSSsub: ios_$read_ptr - read from ioname "user_input"

"reading" 2
     AMLsub: ipc_$cutoff - inhibit event channel for reading
     AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)

"readsync" 1
     SSSsub: ios_$readsync - set read synchronization mode for an ioname

"ready" 3
     SSScmd: ready (rdy) - print listener ready message
     SSScmd: ready_off (rdf) - turn off listener ready messages
     SSScmd: ready_on (rdn) - turn on listener ready messages

"ready_off" 1
     SSScmd: ready_off (rdf) - turn off listener ready messages

"ready_on" 1
     SSScmd: ready_on (rdn) - turn on listener ready messages

"receive" 1
     remark: an event channel is used to receive inter-process messages and wakeups

"reconnect" 1
     AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)

"record" 5
     remark: a directory may have a record quota, which limits the number of pages which may be charged to it
     remark: if the record quota of a directory is 0, its pages are charged to its parent directory
     SSScmd: getquota (gq) - print out record quota and current use for given directory
     SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
     SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch

"records" 1
     remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them

"redef" 1
```

```
     SSSsub: area_$redef - change the length of an area

"reduce" 1
     SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame

"reference" 21
     remark: the Multics Programmer's Manual (MPM) is the reference manual for the system
     remark: a link entry gives the path name of some other entry; it is like an indirect reference
     remark: the KST (known segment table) is a per-process table of reference names and segment numbers
     remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
     remark: MPM RefData 3.4 describes the linkage section and inter-segment references
     SSS_af: get_pathname (gpn) - return absolute path name for given reference name
     SSScmd: initiate (in) - make a file known by a given reference name
     SSScmd: terminate (tm) - terminate a reference name
     SSScmd: terminate_refname (tmr) - terminate a segment given by reference name
     SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
     SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
     SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
     SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
     SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
     SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
     SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
     SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
     SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown
     SSSsub: ios_$seek - set reference pointers associated with an ioname
     SSSsub: ios_$tell - return value of reference pointer associated with an ioname
     SSSsub: term_$refname - same as term_ for segment given by reference name

"refname" 1
     SSSsub: term_$refname - same as term_ for segment given by reference name

"relate" 1
     SSSsub: hcs_$set_relatesw - set relate switch in branch entry

"relative" 2
     remark: a relative path name is relative to the current working directory; it begins with "<" or a name
     SSSsub: expand_path_ - expand relative path name into absolute path name

"release" 1
     SSScmd: release - release machine conditions saved by "hold"

"remark" 65
     remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645
     remark: the Multics Programmer's Manual (MPM) is the reference manual for the system
     remark: a user normally controls his Multics process from a typewriter terminal (console)
     remark: a "process" is a locus of control within an instruction sequence
     remark: a process has a two-dimensional address space consisting of segments
     remark: a "segment" is an addressable collection of words which has names and access attributes
     remark: segments are divided into 1024-word pages which are brought into core by the file system on demand
     remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
     remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
     remark: a link entry gives the path name of some other entry; it is like an indirect reference
     remark: every known segment in a process is associated with some file or directory branch
     remark: each process has a process directory, where per-process segments are kept
     remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry
     remark: an absolute path name begins with ">" and leads from the root directory to some entry
     remark: a relative path name is relative to the current working directory; it begins with "<" or a name
     remark: the KST (known segment table) is a per-process table of reference names and segment numbers
     remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
```

remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signalled
remark: the segments addressable by a process are grouped into rings of protection
remark: a user can enter a lower-numbered protection ring only through a "gate" segment
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch
remark: each item on an ACL gives an access name and the access mode permitted
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
remark: a common access control list (CACL) is a list of items giving default access for branches in a directory
remark: a directory may have a record quota, which limits the number of pages which may be charged to it
remark: if the record quota of a directory is 0, its pages are charged to its parent directory
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
remark: an "ioname" is the name of an input-output stream or device channel
remark: users make calls to the input-output switch "ios_" to read and write on ionames
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
remark: an event channel is used to receive inter-process messages and wakeups
remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
remark: "signal_" signals the "unclaimed_signal" condition if it cannot find a handler for a condition
remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 1.1 describes the login and logout commands and system messages
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 1.6 describes punch-card input and control cards
remark: MPM RefData 2.1 describes local conventions for Multics users
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
remark: MPM RefData 3.4 describes the linkage section and inter-segment references
remark: MPM RefData 3.5 describes magnetic tape formats
remark: MPM RefData 3.7 describes punched card codes
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines
remark: SSS commands are marked "SSScmd" in this index; consult the SSS section of the MPM
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM

"remove" 1
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment

"rename" 1
SSScmd: rename (rn) - change an entry name for a file

"reorder_archive" 1
SSScmd: reorder_archive - change order of component files in an archive file

"repeat" 1

```
     AMLcmd: return_to$repeat_call - return to given stack frame and repeat call

"repeat_call" 1
     AMLcmd: return_to$repeat_call - return to given stack frame and repeat call

"replace" 1
     SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch

"request" 22
     remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
     SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
     SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch
     AMLcmd: eplbsa - request delayed assembly of programs in the eplbsa language
     SSScmd: logout - request process destruction and session termination
     SSScmd: new_proc - request user control to create a new user process
     SSScmd: pl1d: request delayed compile of programs in the pl1 language
     SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds
     SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
     SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
     SSSsub: hcs_$assign - request GIM to assign and initialize device channel
     SSSsub: hcs_$list_change - request GIM to change DCW list
     SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
     SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
     SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds
     SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
     SSSsub: hcs_$unassign - request GIM to un-assign a device channel
     SSSsub: ios_$order - issue request to outer module associated with an ioname
     AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channels
     AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
     AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels
     AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channels

"resetread" 1
     SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname

"resetwrite" 1
     SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname

"respect" 1
     SSSsub: hcs_$fs_get_mode - return access of user with respect to segment

"restart" 1
     SSScmd: start - restart execution at saved machine conditions

"return" 75
     remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
     remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
     SSS_af: get_pathname (gpn) - return absolute path name for given reference name
     SSS_af: pd - return absolute path name of process directory
     AMLcmd: return_to - return to a given stack frame
     AMLcmd: return_to$repeat_call - return to given stack frame and repeat call
     SSS_af: wd - return absolute path name of current working directory
     SSSsub: alloc_ - allocate storage in given area, return pointer
     SSSsub: clock_ - return calendar clock time in microseconds
     SSSsub: com_err_$check_fs_errcode_ - return character string for system error code
     SSSsub: cu_$arg_count - return number of arguments of calling procedure
     SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
     SSSsub: cu_$arg_ptr - return pointer to character-string argument
     SSSsub: cu_$level_get - return current validation level
```

SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
SSSsub: get_group_id_ - return user's access control name
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star
SSSsub: get_pdir_ - return absolute path name of process directory
SSSsub: get_process_id_ - return user process id
SSSsub: get_ring_ - return the current ring number
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_$fs_get_mode - return access of user with respect to segment
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_$get_cur_status - return SIM position in DCW list
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_$get_page_trace - return information about last 200 page faults
SSSsub: hcs_$get_status - return SIM status for device channel
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
SSSsub: hcs_$status - return information from directory entry about a branch or link
SSSsub: hcs_$status_long - return all information from directory entry about a branch or link
SSSsub: hcs_$status_minf - return bit count and type for a branch or link
SSSsub: hcs_$status_mins - return bit count and type for a segment
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used
SSSsub: ioa_$rs - format a message and return character string
SSSsub: ioa_$rsnnl - format a message and return character string, no newline
SSSsub: ios_$getdelim - return current read delimiters for ioname
SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname
SSSsub: ios_$getsize - return current element size for an ioname
SSSsub: ios_$tell - return value of reference pointer associated with an ioname
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
AMLsub: ipc_$read_ev_chn - return event message from an event channel
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
AMLsub: random_$exponential - return random number from exponential distribution with mean 1
AMLsub: random_$exponential_seq - return array of random numbers from exponential distribution
AMLsub: random_$normal - return random number from normal distribution
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_seq - return array of random numbers from normal distribution
AMLsub: random_$random_get_seed - return current seed for random numbers
AMLsub: random_$normal_ant_seq - return array of random numbers from antithetic normal distribution
AMLsub: random_$uniform - return random number from uniform distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random numbers from uniform distribution

```
     SSSsub: unique_bits_ - return unique bit string
     AMLsub: whoami_ - return user's name and project as given at login

"return_to" 2
     AMLcmd: return_to - return to a given stack frame
     AMLcmd: return_to$repeat_call - return to given stack frame and repeat call

"reverse_index" 1
     AMLsub: reverse_index - scan character string from right to left for given character

"reversion_" 1
     SSSsub: reversion_ - revert a condition handler

"revert" 1
     SSSsub: reversion_ - revert a condition handler

"rewa" 1
     remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null

"right" 1
     AMLsub: reverse_index - scan character string from right to left for given character

"rights" 1
     remark: an access control list (ACL) is a per-branch list of items giving access rights to the branch

"ring" 9
     remark: the segments addressable by a process are grouped into rings of protection
     remark: a user can enter a lower-numbered protection ring only through a "gate" segment
     remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
     remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
     remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
     SSSsub: get_ring_ - return the current ring number
     SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
     SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
     SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number

"rn" 1
     SSScmd: rename (rn) - change an entry name for a file

"rndshftb" 1
     AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n

"rndshftb36" 1
     AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n

"root" 1
     remark: an absolute path name begins with ">" and leads from the root directory to some entry

"round" 4
     AMLsub: roundb - round a number to next multiple of 2**n
     AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
     AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
     AMLsub: roundb$rounib36 - round a number to multiple of 36, round to multiple of 2**n

"roundb" 4
     AMLsub: roundb - round a number to next multiple of 2**n
     AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
     AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
     AMLsub: roundb$rounib36 - round a number to multiple of 36, round to multiple of 2**n
```

"roundb36" 1
    AMLsub: roundb$roundb36 - round a number to multiple of 36, round to multiple of 2**n

"rs" 1
    SSSsub: ioa_$rs - format a message and return character string

"rsnnl" 1
    SSSsub: ioa_$rsnnl - format a message and return character string, no newline

"rules" 1
    remark: MPM RefData 2.7 describes the system standard search rules used by the linker

"runoff" 1
    AMLcmd: runoff - format a character file for printing

Index for Multics - 93 words beginning with s:

"SSS" 5
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines
remark: SSS commands are marked "SSScmd" in this index; consult the SSS section of the MPM
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS

"SSS_af" 4
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
SSS_af: get_pathname (gpn) - return absolute path name for given reference name
SSS_af: pd - return absolute path name of process directory
SSS_af: wd - return absolute path name of current working directory

"SSScmd" 79
remark: SSS commands are marked "SSScmd" in this index; consult the SSS section of the MPM
SSScmd: addname (an) - add additional entry names to file branch
SSScmd: archive (ac) - combine several files into an archive file
SSScmd: archive_sort (as) - sort an archive file on component file names
SSScmd: basic - compile and execute a program in the basic language
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language
SSScmd: calc - desk calculator with memory for functions and expressions
SSScmd: changewdir (cwd) - change current working directory
SSScmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"
SSScmd: copy (cp) - copy a file
SSScmd: create (cr) - create a file branch
SSScmd: createdir (cd) - create a directory branch
SSScmd: decam - desk calculator with memory
SSScmd: delete (dl) - delete a file
SSScmd: deleteacl (da) - delete item from an ACL
SSScmd: deletecacl (dc) - delete item from a CACL
SSScmd: deletedir (dd) - delete a directory branch and all files contained in it
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
SSScmd: deletename (dn) - delete entry name from a file branch
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch
SSScmd: edm - context editor for character files
SSScmd: endfile - close fortran files
SSScmd: exec_com (ec) - execute a series of command lines contained in a file
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console
SSScmd: fortran - compile programs in the fortran language
SSScmd: get_com_line (gcl) - print out current expanded command line length
SSScmd: getquota (gq) - print out record quota and current use for given directory
SSScmd: global (gb) - execute a command in every directory inferior to a given directory
SSScmd: help - print out system information files
SSScmd: hold - save machine conditions after error
SSScmd: how_many_users (hmu) - print out current number of users and maximum number
SSScmd: initiate (in) - make a file known by a given reference name
SSScmd: iocall - call "ios_" with given arguments
SSScmd: iomode - set character conversion mode for an ioname
SSScmd: line_length (ll) - set maximum line length for an ioname
SSScmd: link (lk) - create a link entry
SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSScmd: listacl (la) - print out ACL for a file
SSScmd: listcacl (lc) - print out CACL for a directory
SSScmd: listnames (ln) - print out entry names for files in a directory

SSScmd: listotals (lt) - print out entry counts and use for a directory
SSScmd: login - identify user by name and project to system control and create user process
SSScmd: logout - request process destruction and session termination
SSScmd: moveb (mv) - move a file to another directory
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
SSScmd: new_proc - request user control to create a new user process
SSScmd: page_trace (pgt) - print out information about last 200 page faults
SSScmd: pl1 - compile programs in the pl1 language
SSScmd: pl1d: request delayed compile of programs in the pl1 language
SSScmd: print (pr) - print a character file
SSScmd: print_attach_table (pat) - print ioname attachments for process
SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
SSScmd: print_object_map (pom) - print object map for an object segment
SSScmd: printhomedir (phd) - print absolute path name of home directory
SSScmd: printwdir (pwd) - print absolute path name of current working directory
SSScmd: probe - interactive debugging aid
SSScmd: program_interrupt (pi) - signal the condition "program_interrupt"
SSScmd: ready (rdy) - print listener ready message
SSScmd: ready_off (rdf) - turn off listener ready messages
SSScmd: ready_on (rdn) - turn on listener ready messages
SSScmd: release - release machine conditions saved by "hold"
SSScmd: rename (rn) - change an entry name for a file
SSScmd: reorder_archive - change order of component files in an archive file
SSScmd: set_bit_count (sbc) - set bit count for a file branch
SSScmd: set_com_line (scl) - set maximum expanded command line length
SSScmd: setacl (sa) - add item to ACL for a file
SSScmd: setcacl (sc) - add item to CACL for a directory
SSScmd: sethomedir (shd) - set path name of home directory
SSScmd: start - restart execution at saved machine conditions
SSScmd: status (st) - print out status information from directory entry for a branch
SSScmd: terminate (tm) - terminate a reference name
SSScmd: terminate_refname (tmr) - terminate a segment given by reference name
SSScmd: terminate_segno (tms) - terminate a segment given by segment number
SSScmd: unlink (ul) - delete a link entry from a directory
SSScmd: who - print out names of users on system
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSScmd: who$long - print out names and project id's for users on system

"SSSsub" 160
remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds
SSSsub: acm_$sleep_interval - block process for time interval in microseconds
SSSsub: acm_$sleep_time - block process until time given in microseconds
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds
SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
SSSsub: alloc_ - allocate storage in given area, return pointer
SSSsub: area_ - initialize an area
SSSsub: area_$redef - change the length of an area
SSSsub: check_star_ - check whether path name contains a star
SSSsub: clock_ - return calendar clock time in microseconds
SSSsub: com_err_ - format system error message and write on "user_output"
SSSsub: com_err_$check_fs_errcode_ - return character string for system error code
SSSsub: com_err_$notify - format non-system error message and write on "user_output"
SSSsub: condition_ - establish handler for a condition
SSSsub: cu_$arg_count - return number of arguments of calling procedure
SSSsub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
SSSsub: cu_$arg_ptr - return pointer to character-string argument

SSSsub: cu_$cl - call current listener to re-enter command level
SSSsub: cu_$cp - call current command processor to execute command line
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame
SSSsub: cu_$level_get - return current validation level
SSSsub: cu_$level_set - set validation level
SSSsub: cu_$ptr_call - call procedure given by pointer with given arguments
SSSsub: cu_$set_cl - define current listener
SSSsub: cu_$set_cp - define current command processor
SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame
SSSsub: cv_bin_ - convert binary to character string, any base
SSSsub: cv_bin_$dec - convert binary to character string, base 10
SSSsub: cv_bin_$oct - convert binary to character string, base 8
SSSsub: cv_dec_ - convert character string to binary, base 10
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSsub: cv_oct_ - convert character string to binary, base 8
SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
SSSsub: date_time_ - convert calendar clock time to character string date and time
SSSsub: date_time_$fstime - convert file system time to character string date and time
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
SSSsub: equal_ - expand entry name which may contain equal signs
SSSsub: expand_path_ - expand relative path name into absolute path name
SSSsub: freen_ - free allocated storage in an area
SSSsub: get_at_entry_ - return outer module, attached ioname, and mode for given ioname
SSSsub: get_group_id_ - return user's access control name
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star
SSSsub: get_pdir_ - return absolute path name of process directory
SSSsub: get_process_id_ - return user process id
SSSsub: get_ring_ - return the current ring number
SSSsub: hcs_$acl_add - add or change items on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$append_link - create a link entry in parent directory
SSSsub: hcs_$assign - request GIM to assign and initialize device channel
SSSsub: hcs_$assign_linkage - allocate storage in linkage section
SSSsub: hcs_$chname_file - add and delete entry names on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry names on branch or link entry
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_$del_dir_tree - delete all branches inferior to a directory
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_$fs_get_mode - return access of user with respect to segment
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_$fs_move_file - move contents from one file branch to another
SSSsub: hcs_$fs_move_seg - move contents from one segment to another
SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_$fs_search_set_wdir - change current working directory
SSSsub: hcs_$get_cur_status - return GIM position in DCW list
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment

SSSsub: hcs_$get_page_trace - return information about last 200 page faults
SSSsub: hcs_$get_status - return GIM status for device channel
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry
SSSsub: hcs_$set_copysw - set copy switch in branch entry
SSSsub: hcs_$set_relatesw - set relate switch in branch entry
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
SSSsub: hcs_$star_ - search a directory for entry names matching template
SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information
SSSsub: hcs_$status - return information from directory entry about a branch or link
SSSsub: hcs_$status_long - return all information from directory entry about a branch or link
SSSsub: hcs_$status_minf - return bit count and type for a branch or link
SSSsub: hcs_$status_mins - return bit count and type for a segment
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown
SSSsub: hcs_$truncate_file - truncate file branch from a given offset
SSSsub: hcs_$truncate_seg - truncate segment from a given offset
SSSsub: hcs_$unassign - request GIM to un-assign a device channel
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used
SSSsub: ioa_ - format a message and write it on "user_output"
SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
SSSsub: ioa_$rs - format a message and return character string
SSSsub: ioa_$rsnnl - format a message and return character string, no newline
SSSsub: ios_$abort - abort outstanding transactions for an ioname
SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSsub: ios_$changemode - change mode of attachment of an ioname
SSSsub: ios_$detach - un-attach one ioname from another ioname
SSSsub: ios_$getdelim - return current read delimiters for ioname
SSSsub: ios_$getmode - return binary encoding of mode of attachment for ioname
SSSsub: ios_$getsize - return current element size for an ioname
SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
SSSsub: ios_$order - issue request to outer module associated with an ioname
SSSsub: ios_$read - read from an ioname
SSSsub: ios_$read_ptr - read from ioname "user_input"
SSSsub: ios_$readsync - set read synchronization mode for an ioname
SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname
SSSsub: ios_$seek - set reference pointers associated with an ioname
SSSsub: ios_$setdelim - set current read delimiters for an ioname
SSSsub: ios_$setsize - set current element size for an ioname
SSSsub: ios_$tell - return value of reference pointer associated with an ioname
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname
SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
SSSsub: ios_$write - write on an ioname

SSSsub: ios_$write_ptr - write on ioname "user_output"
SSSsub: ios_$writesync - set write synchronization mode for an ioname
SSSsub: make_obj_map_ - write an object map into object segment
SSSsub: move_ - move a block of words from one place to another (given by pointers)
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: parse_file_$parse_file_set_break - declare break characters to parse_file_
SSSsub: parse_file_$parse_file_unset_break - undeclare break characters for parse_file_
SSSsub: reversion_ - revert a condition handler
SSSsub: signal_ - signal the occurrence of a condition, call its handler
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
SSSsub: term_$refname - same as term_ for segment given by reference name
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer
SSSsub: ti_$findata - turn scratch segment into data segment
SSSsub: ti_$finobj - turn scratch segment into object segment
SSSsub: ti_$getseg - set up a scratch segment
SSSsub: unique_bits_ - return unique bit string
SSSsub: unique_chars_ - convert bit string into unique character string

"Service" 1
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines

"Standard" 1
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines

"System" 1
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines

"sa" 1
SSScmd: setacl (sa) - add item to ACL for a file

"safety" 1
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list

"same" 3
SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
SSSsub: term_$refname - same as term_ for segment given by reference name
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer

"satisfied" 1
remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signalled

"save" 5
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2,3)
remark: MPM RefData 2,3 gives the formats of the stack and argument lists and the call-save-return sequences
SSScmd: hold - save machine conditions after error
SSScmd: release - release machine conditions saved by "hold"
SSScmd: start - restart execution at saved machine conditions

"sbc" 1
SSScmd: set_bit_count (sbc) - set bit count for a file branch

"sc" 1
SSScmd: setcacl (sc) - add item to CACL for a directory

"scan" 3
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
AMLsub: reverse_index - scan character string from right to left for given character

"scl" 1
SSScmd: set_com_line (scl) - set maximum expanded command line length

"scratch" 3
SSSsub: ti_$findata - turn scratch segment into data segment
SSSsub: ti_$finobj - turn scratch segment into object segment
SSSsub: ti_$getseg - set up a scratch segment

"search" 5
remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_$star_ - search a directory for entry names matching template
SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information

"second" 2
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds

"section" 12
remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
remark: MPM RefData 3.4 describes the linkage section and inter-segment references
remark: SSS commands are marked "SSScmd" in this index; consult the SSS section of the MPM
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
SSSsub: hcs_$assign_linkage - allocate storage in linkage section
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment

"see" 3
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: the 128 ASCII characters are coded as 7-bit characters in 9-bit bytes (see MPM RefData 3.1)

"seed" 2
AMLsub: random_$random_get_seed - return current seed for random numbers
AMLsub: random_$random_set_seed - set seed for random numbers

"seek" 1
SSSsub: ios_$seek - set reference pointers associated with an ioname

"seg_ptr" 1
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer

"segment" 50
remark: a process has a two-dimensional address space consisting of segments
remark: a "segment" is an addressable collection of words which has names and access attributes
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand
remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
remark: every known segment in a process is associated with some file or directory branch

```
     remark: each process has a process directory, where per-process segments are kept
     remark: the KST (known segment table) is a per-process table of reference names and segment numbers
     remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section
     remark: the segments addressable by a process are grouped into rings of protection
     remark: a user can enter a lower-numbered protection ring only through a "gate" segment
     remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
     remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
     remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic
     remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
     remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
     remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
     remark: MPM RefData 3.4 describes the linkage section and inter-segment references
     SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segments
     AMLcmd: ctss_aarchv - extract all component files from a CTSS ASCII archive segment
     SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
     SSScmd: print_object_map (pom) - print object map for an object segment
     SSScmd: terminate_refname (tmr) - terminate a segment given by reference name
     SSScmd: terminate_segno (tms) - terminate a segment given by segment number
     SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
     SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
     SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
     SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
     SSSsub: hcs_$fs_get_mode - return access of user with respect to segment
     SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
     SSSsub: hcs_$fs_get_ref_name - return reference names for a segment from KST
     SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
     SSSsub: hcs_$fs_move_seg - move contents from one segment to another
     SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
     SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
     SSSsub: hcs_$status_mins - return bit count and type for a segment
     SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
     SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
     SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
     SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown
     SSSsub: hcs_$truncate_seg - truncate segment from a given offset
     SSSsub: make_obj_map_ - write an object map into object segment
     SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
     SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment
     SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
     SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
     SSSsub: term_$refname - same as term_ for segment given by reference name
     SSSsub: term_$seg_ptr - same as term_ for segment given by pointer
     SSSsub: ti_$findata - turn scratch segment into data segment
     SSSsub: ti_$finobj - turn scratch segment into object segment
     SSSsub: ti_$getseg - set up a scratch segment

"send" 2
     AMLcmd: mail (ml) - send a file to another user's mailbox
     AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id

"sent" 1
     remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel

"separated" 1
     remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry

"sequence" 3
     remark: a "process" is a locus of control within an instruction sequence
     remark: a "path name" is a sequence of entry names separated by ">" which specifies an entry
```

remark: MPM RefData 2,3 gives the formats of the stack and argument lists and the call-save-return sequences

"series" 1
SSScmd: exec_com (ec) - execute a series of command lines contained in a file

"service" 1
remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645

"session" 1
SSScmd: logout - request process destruction and session termination

"set" 23
AMLcmd: adjust - set bit count on file branch
AMLcmd: adjust$block - set bit count on file branch, start from current length
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
AMLcmd: entry_usage$clear - set entrypoint usage counts to 0
SSScmd: iomode - set character conversion mode for an ioname
SSScmd: line_length (ll) - set maximum line length for an ioname
SSScmd: set_bit_count (sbc) - set bit count for a file branch
SSScmd: set_com_line (scl) - set maximum expanded command line length
SSScmd: sethomedir (shd) - set path name of home directory
SSSsub: cu_$level_set - set validation level
SSSsub: hcs_$list_size - set size of SIM DCW list for device channel
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry
SSSsub: hcs_$set_copysw - set copy switch in branch entry
SSSsub: hcs_$set_relatesw - set relate switch in branch entry
SSSsub: ios_$readsync - set read synchronization mode for an ioname
SSSsub: ios_$seek - set reference pointers associated with an ioname
SSSsub: ios_$setdelim - set current read delimiters for an ioname
SSSsub: ios_$setsize - set current element size for an ioname
SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
SSSsub: ios_$writesync - set write synchronization mode for an ioname
AMLsub: random_$ranion_set_seed - set seed for random numbers
SSSsub: ti_$getseg - set up a scratch segment

"set_alarm" 1
SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds

"set_bc" 1
SSSsub: hcs_$set_bc - set bit count in file branch entry

"set_bit_count" 1
SSScmd: set_bit_count (sbc) - set bit count for a file branch

"set_call_prior" 1
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels

"set_cl" 1
SSSsub: cu_$set_cl - define current listener

"set_com_line" 1
SSScmd: set_com_line (scl) - set maximum expanded command line length

"set_consistsw" 1
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry

"set_copysw" 1
SSSsub: hcs_$set_copysw - set copy switch in branch entry

```
"set_cp" 1
     SSSsub: cu_$set_cp - define current command processor

"set_relatesw" 1
     SSSsub: hcs_$set_relatesw - set relate switch in branch entry

"set_timer" 1
     SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval

"set_wait_prior" 1
     AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels

"setacl" 1
     SSScmd: setacl (sa) - add item to ACL for a file

"setcacl" 1
     SSScmd: setcacl (sc) - add item to CACL for a directory

"setdelim" 1
     SSSsub: ios_$setdelim - set current read delimiters for an ioname

"sethomedir" 1
     SSScmd: sethomedir (shd) - set path name of home directory

"sets" 1
     remark: the linker searches for the called segment and entrypoint and sets a pointer in the linkage section

"setsize" 1
     SSSsub: ios_$setsize - set current element size for an ioname

"several" 1
     SSScmd: archive (ac) - combine several files into an archive file

"sharing" 1
     remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645

"shd" 1
     SSScmd: sethomedir (shd) - set path name of home directory

"shrink_stack_frame" 1
     SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame

"sign" 1
     SSSsub: equal_ - expand entry name which may contain equal signs

"signal" 7
     remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signalled
     remark: "signal_" signals the "unclaimed_signal" condition if it cannot find a handler for a condition
     remark: "quit" is signalled after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)
     remark: MPM RefData 2,2,1 lists all condition names which the system may signal
     SSScmd: program_interrupt (pi) - signal the condition "program_interrupt"
     SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
     SSSsub: signal_ - signal the occurrence of a condition, call its handler

"signal_" 2
     remark: "signal_" signals the "unclaimed_signal" condition if it cannot find a handler for a condition
     SSSsub: signal_ - signal the occurrence of a condition, call its handler
```

"single" 1
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch

"size" 6
SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame
SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame
SSSsub: hcs_$list_size - set size of SIM DCW list for device channel
SSSsub: ios_$getsize - return current element size for an ioname
SSSsub: ios_$setsize - set current element size for an ioname

"sleep_int_secs" 1
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds

"sleep_interval" 1
SSSsub: acm_$sleep_interval - block process for time interval in microseconds

"sleep_time" 1
SSSsub: acm_$sleep_time - block process until time given in microseconds

"some" 4
remark: a link entry gives the path name of some other entry; it is like an indirect reference
remark: every known segment in a process is associated with some file or directory branch
remark: an absolute path name begins with ">" and leads from the root directory to some entry
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel

"sort" 1
SSScmd: archive_sort (as) - sort an archive file on component file names

"space" 1
remark: a process has a two-dimensional address space consisting of segments

"specifies" 1
remark: a "Path name" is a sequence of entry names separated by ">" which specifies an entry

"speed" 2
SSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSScmd: dpunch (dpn) - request delayed punching of a file on high-speed punch

"st" 1
SSScmd: status (st) - print out status information from directory entry for a branch

"stack" 8
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: MPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
AMLcmd: return_to - return to a given stack frame
AMLcmd: return_to$repeat_call - return to given stack frame and repeat call
SSSsub: cu_$grow_stack_frame - increase size of calling procedure's stack frame
SSSsub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame

"stack_frame_ptr" 1
SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure

"stack_frame_size" 1
SSSsub: cu_$stack_frame_size - return size of calling procedure's stack frame

"standard" 1

```
     remark: MPM RefData 2.7 describes the system standard search rules used by the linker

"star" 2
     SSSsub: check_star_ - check whether path name contains a star
     SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star

"star_" 1
     SSSsub: hcs_$star_ - search a directory for entry names matching template

"star_list_" 1
     SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information

"start" 5
     AMLcmd: adjust$block - set bit count on file branch, start from current length
     AMLcmd: adjust$block_test - check bit count on file branch, start from current length
     AMLcmd: adjust$test_block - check bit count on file branch, start from current length
     SSScmd: start - restart execution at saved machine conditions
     SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list

"status" 4
     SSScmd: status (st) - print out status information from directory entry for a branch
     SSSsub: hcs_$get_status - return GIM status for device channel
     SSSsub: hcs_$status - return information from directory entry about a branch or link
     SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname

"status_long" 1
     SSSsub: hcs_$status_long - return all information from directory entry about a branch or link

"status_minf" 1
     SSSsub: hcs_$status_minf - return bit count and type for a branch or link

"status_mins" 1
     SSSsub: hcs_$status_mins - return bit count and type for a segment

"stop" 1
     SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list

"storage" 3
     SSSsub: alloc_ - allocate storage in given area, return pointer
     SSSsub: freen_ - free allocated storage in an area
     SSSsub: hcs_$assign_linkage - allocate storage in linkage section

"stream" 1
     remark: an "ioname" is the name of an input-output stream or device channel

"string" 18
     remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
     AMLcmd: ps - print argument string
     SSSsub: com_err_$check_fs_errcode_ - return character string for system error code
     SSSsub: cu_$arg_ptr - return pointer to character-string argument
     SSSsub: cv_bin_ - convert binary to character string, any base
     SSSsub: cv_bin_$dec - convert binary to character string, base 10
     SSSsub: cv_bin_$oct - convert binary to character string, base 8
     SSSsub: cv_dec_ - convert character string to binary, base 10
     SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
     SSSsub: cv_oct_ - convert character string to binary, base 8
     SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
     SSSsub: date_time_ - convert calendar clock time to character string date and time
     SSSsub: date_time_$fstime - convert file system time to character string date and time
```

SSSsub: ioa_$rs - format a message and return character string
SSSsub: ioa_$rsnnl - format a message and return character string, no newline
AMLsub: reverse_index - scan character string from right to left for given character
SSSsub: unique_bits_ - return unique bit string
SSSsub: unique_chars_ - convert bit string into unique character string

"subroutine" 3
remark: the Standard Service System (SSS) is a group of carefully checked and efficient commands and subroutines
remark: SSS subroutines are marked "SSSsub" in this index; consult the SSS section of the MPM
remark: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS

"subroutines" 1
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM

"supervisor" 1
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor

"switch" 4
remark: users make calls to the input-output switch "ios_" to read and write on ionames
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry
SSSsub: hcs_$set_copysw - set copy switch in branch entry
SSSsub: hcs_$set_relatesw - set relate switch in branch entry

"symbol" 6
AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: debug (db) - symbolic interactive debugging aid
AMLcmd: printsymtab (pst) - print symbol table for a pl1 program
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol

"symbolic" 1
remark: inter-segment references are of the form "segmentname$entrypoint"; both parts are symbolic

"synchronization" 3
SSSsub: ios_$readsync - set read synchronization mode for an ioname
SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
SSSsub: ios_$writesync - set write synchronization mode for an ioname

"system" 19
remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645
remark: the Multics Programmer's Manual (MPM) is the reference manual for the system
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
remark: MPM RefData 1.3 gives all the system error codes
remark: MPM RefData 1.1 describes the login and logout commands and system messages
remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
remark: MPM RefData 2.2.1 lists all condition names which the system may signal
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
SSScmd: help - print out system information files
SSScmd: login - identify user by name and project to system control and create user process
SSScmd: who - print out names of users on system
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSScmd: who$long - print out names and project id's for users on system
SSSsub: com_err_ - format system error message and write on "user_output"
SSSsub: com_err_$check_fs_errcode_ - return character string for system error code
SSSsub: com_err_$notify - format non-system error message and write on "user_output"
SSSsub: date_time_$fstime - convert file system time to character string date and time

"Teletype" 1
remark: "quit" is signalled after a "quit signal" (ATIN on 2741, INTERRUPT on model 37 Teletype)

"tab" 1
AMLcmd: pht - print horizontal tab

"table" 4
remark: a "directory" is a system-maintained table; each entry contains the names and attributes of a branch or link
remark: the KST (known segment table) is a per-process table of reference names and segment numbers
AMLcmd: createSymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: printsymtab (pst) - print symbol table for a pl1 program

"tag" 1
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star

"tag_star" 1
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star

"tape" 1
remark: MPM RefData 3,5 describes magnetic tape formats

"tell" 1
SSSsub: ios_$tell - return value of reference pointer associated with an ionane

"template" 2
SSSsub: hcs_$star_ - search a directory for entry names matching template
SSSsub: hcs_$star_list_ - search a directory for entry names matching template, return long information

"term_" 4
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
SSSsub: term_$refname - same as term_ for segment given by reference name
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer

"terminal" 3
remark: a user normally controls his Multics process from a typewriter terminal (console)
remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
SSScmd: bsys - interactive editor and terminal interface for programming in the basic language

"terminate" 9
SSScmd: terminate (tm) - terminate a reference name
SSScmd: terminate_refname (tmr) - terminate a segment given by reference name
SSScmd: terminate_segno (tms) - terminate a segment given by segment number
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown
SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
SSSsub: term_$nomakeunknown - same as term_ for segment given by pointer, but do not terminate

"terminate_file" 1
SSSsub: hcs_$terminate_file - terminate all reference names for branch from KST, make segment unknown

"terminate_name" 1
SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST

"terminate_noname" 1
SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST

"terminate_refname" 1
SSScmd: terminate_refname (tmr) - terminate a segment given by reference name

"terminate_seg" 1
SSSsub: hcs_$terminate_seg - terminate all reference names for a segment from KST, make segment unknown

"terminate_segno" 1
SSScmd: terminate_segno (tms) - terminate a segment given by segment number

"termination" 1
SSScmd: logout - request process destruction and session termination

"test" 1
AMLcmd: adjust$test - check bit count on file branch

"test_block" 1
AMLcmd: adjust$test_block - check bit count on file branch, start from current length

"text" 2
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
SSSsub: decode_object_ - return pointers to text, linkage, and symbol section of object segment

"through" 1
remark: a user can enter a lower-numbered protection ring only through a "gate" segment

"ti_" 3
SSSsub: ti_$findata - turn scratch segment into data segment
SSSsub: ti_$finobj - turn scratch segment into object segment
SSSsub: ti_$getseg - set up a scratch segment

"time" 11
remark: Multics (multiplexed information and computing service) is a time-sharing system on the GE645
SSSsub: acm_$sleep_int_secs - block process for time interval in seconds
SSSsub: acm_$sleep_interval - block process for time interval in microseconds
SSSsub: acm_$sleep_time - block process until time given in microseconds
SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
SSSsub: clock_ - return calendar clock time in microseconds
SSSsub: date_time_ - convert calendar clock time to character string date and time
SSSsub: date_time_$fstime - convert file system time to character string date and time
SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used

"tm" 1
SSScmd: terminate (tm) - terminate a reference name

"tmr" 1
SSScmd: terminate_refname (tmr) - terminate a segment given by reference name

"tms" 1
SSScmd: terminate_segno (tms) - terminate a segment given by segment number

"transaction" 3
SSSsub: ios_$abort - abort outstanding transactions for an ioname
SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname

```
"truncate" 2
     SSSsub: hcs_$truncate_file - truncate file branch from a given offset
     SSSsub: hcs_$truncate_seg - truncate segment from a given offset

"truncate_file" 1
     SSSsub: hcs_$truncate_file - truncate file branch from a given offset

"truncate_seg" 1
     SSSsub: hcs_$truncate_seg - truncate segment from a given offset

"turn" 4
     SSScmd: ready_off (rif) - turn off listener ready messages
     SSScmd: ready_on (rin) - turn on listener ready messages
     SSSsub: ti_$findata - turn scratch segment into data segment
     SSSsub: ti_$finobj - turn scratch segment into object segment

"two" 2
     remark: a process has a two-dimensional address space consisting of segments
     AMLcmd: compare_ascii (cpa) - compare two character files

"type" 5
     remark: MPM RefData 1.4 describes the input-output system and device and ourter module types
     SSSsub: hcs_$status_minf - return bit count and type for a branch or link
     SSSsub: hcs_$status_mins - return bit count and type for a segment
     AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
     AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)

"typewriter" 1
     remark: a user normally controls his Multics process from a typewriter terminal (console)

"typing" 1
     remark: MPM RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters
```

SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname

"upstate" 1
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname

"usage" 2
AMLcmd: entry_usage - print out usage counts for entrypoints
AMLcmd: entry_usage$clear - set entrypoint usage counts to 0

"usage_values" 1
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used

"use" 4
remark: MPM RefData 2.7 describes the system standard search rules used by the linker
SSScmd: getquota (gq) - print out record quota and current use for given directory
SSScmd: listotals (lt) - print out entry counts and use for a directory
SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch

"used" 3
remark: an event channel is used to receive inter-process messages and wakeups
AMLcmd: createsymtab (cst) - create symbol table files for pl1-language programs (used with "debug")
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used

"user" 22
remark: a user normally controls his Multics process from a typewriter terminal (console)
remark: a user can enter a lower-numbered protection ring only through a "gate" segment
remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardcore) supervisor
remark: an access name is of the form "username.projectname.instance"; any part may be "*"
remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
remark: users make calls to the input-output switch "ios_" to read and write on ionames
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
remark: MPM RefData 2.1 describes local conventions for Multics users
SSScmd: how_many_users (hmu) - print out current number of users and maximum number
SSScmd: login - identify user by name and project to system control and create user process
AMLcmd: mail (ml) - send a file to another user's mailbox
SSScmd: new_proc - request user control to create a new user process
SSScmd: who - print out names of users on system
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSScmd: who$long - print out names and project id's for users on system
SSSsub: get_group_id_ - return user's access control name
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star
SSSsub: get_process_id_ - return user process id
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$fs_get_mode - return access of user with respect to segment
AMLsub: whoami_ - return user's name and project as given at login

"user_dir_dir" 1
remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"

"user_i" 1
SSScmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"

"user_input" 2
SSSsub: ios_$read_ptr - read from ioname "user_input"
AMLsub: read_list_ - read in free-format variables from "user_input"

"user_output" 9

SSScmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console
SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
SSSsub: com_err_ - format system error message and write on "user_output"
SSSsub: com_err_$notify - format non-system error message and write on "user_output"
SSSsub: ioa_ - format a message and write it on "user_output"
SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
SSSsub: ios_$write_ptr - write on ioname "user_output"
AMLsub: write_list_ - write variables on "user_output"

Index for Multics - 4 words beginning with v

"valid" 2
AMLcmd: ascii_check - check all characters of a file for valid ASCII
AMLcmd: ascii_check$archive - check all characters of an archive file for valid ASCII

"validation" 4
SSSsub: cu_$level_get - return current validation level
SSSsub: cu_$level_set - set validation level
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2

"value" 1
SSSsub: ios_$tell - return value of reference pointer associated with an ioname

"variable" 2
AMLsub: read_list_ - read in free-format variables from "user_input"
AMLsub: write_list_ - write variables on "user_output"

Index for Multics - 18 words beginning with w

"wait" 6
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
SSSsub: hcs_$usage_values - return cumulative page waits and CPU time used
SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channels before event-wait channels
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channels before event-call channels

"wakeup" 11
remark: an event channel is used to receive inter-process messages and wakeups
remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel
remark: an event-wait channel records event messages and wakeups; users call "ipc_$block" to wait for them
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds
SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds
SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds
AMLsub: hcs_$block - block process until next wakeup
SSSsub: hcs_$set_alarm - request wakeup at given time in microseconds
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id
AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup

"wakeup_int_secs" 1
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in seconds

"wakeup_interval" 1
SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microseconds

"wakeup_time" 1
SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microseconds

"wd" 1
SSS_af: wd - return absolute path name of current working directory

"when" 2
remark: when one program first calls another, a "linkage fault" occurs and the "linker" is called
remark: an event-call channel has an associated procedure which is called when a wakeup is sent to the channel

"who" 3
SSScmd: who - print out names of users on system
SSScmd: who$admin - print out names, project id's, and process id's for users on system
SSScmd: who$long - print out names and project id's for users on system

"whoami_" 1
AMLsub: whoami_ - return user's name and project as given at login

"within" 1
remark: a "process" is a locus of control within an instruction sequence

"word" 4
remark: a "segment" is an addressable collection of words which has names and access attributes
remark: segments are divided into 1024-word pages which are brought into core by the file system on demand
remark: a DCW (data control word) requests the GIOC to perform some action on a device channel
SSSsub: move_ - move a block of words from one place to another (given by pointers)

"working" 7
remark: a relative path name is relative to the current working directory; it begins with "<" or a name

```
     remark: at login, a user's working directory and home directory are ">user_dir_dir>project>name"
     SSScmd: changewdir (cwd) - change current working directory
     SSScmd: printwdir (pwd) - print absolute path name of current working directory
     SSS_af: wd - return absolute path name of current working directory
     SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
     SSSsub: hcs_$fs_search_set_wdir - change current working directory

"workspace" 2
     SSSsub: ios_$iowait - wait for transactions on workspace asynchronous ioname
     SSSsub: ios_$worksync - set workspace synchronization mode for an ioname

"worksync" 1
     SSSsub: ios_$worksync - set workspace synchronization mode for an ioname

"write" 16
     remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
     remark: users make calls to the input-output switch "ios_" to read and write on ionames
     SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
     SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
     SSSsub: com_err_ - format system error message and write on "user_output"
     SSSsub: com_err_$notify - format non-system error message and write on "user_output"
     SSSsub: ioa_ - format a message and write it on "user_output"
     SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
     SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
     SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
     SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname
     SSSsub: ios_$write - write on an ioname
     SSSsub: ios_$write_ptr - write on ioname "user_output"
     SSSsub: ios_$writesync - set write synchronization mode for an ioname
     SSSsub: make_obj_map_ - write an object map into object segment
     AMLsub: write_list_ - write variables on "user_output"

"write_list_" 1
     AMLsub: write_list_ - write variables on "user_output"

"write_ptr" 1
     SSSsub: ios_$write_ptr - write on ioname "user_output"

"writesync" 1
     SSSsub: ios_$writesync - set write synchronization mode for an ioname
```

```
kwoc: proc;

/* this command produces a KWIC index on any ascii file.
   Usage:
   kwoc file suppress_file

   if "suppress_file" is supplied it will list a
   bunch of words which should be excluded from the index

   THVV 2/70 */

dcl (path, path1) char (168) aligned,                    /* path names */
     ap ptr,                                             /* ptr to arglist */
     pp ptr,                                             /* ptr to input */
     p ptr,                                              /* ptr to suppressfile */
     p1 ptr,                                             /* ptr to wordseg */
     p2 ptr,                                             /* ptr to lineseg */
     bchr char (32) based (ap) unaligned,                /* pickup for args */
     q ptr,                                              /* ptr to word */
     word char (32) aligned,                             /* temporary - word of interest */
     al fixed bin,                                       /* lth of arg */
     ec fixed bin,                                       /* error code */
     bk fixed bin,                                       /* 1 if break char */
     nw fixed bin init (0),                              /* number of ents in wordseg */
     nx fixed bin init (0),                              /* number of ents in linaseg */
     noo fixed bin init (0),                             /* number of suppressed wds */
     alphabet char (52) aligned int static init          /* alphabet */
    ("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"),
     letter char (1) aligned,                            /* letter which begins word */
     NULC char (1) aligned,
     v(0: 26) fixed bin init ((27)0),                    /* pointers to word lists */
     wcount(0: 26) fixed bin init ((27)0),               /* count of real words */
    (oi, ti) fixed bin,                                  /* temps */
     ci fixed bin,                                       /* index of start of word */
     cc fixed bin,                                       /* length of word */
     lno fixed bin,                                      /* line number */
     swap fixed bin init (0),                            /* counter in sort */
     buffer char (300) aligned,                          /* print line buffer */
     1 pchrx based (pp) aligned,                         /* access structure for file */
     2 pchr char (262143),
     out char (32) aligned,                              /* streamname */
     outl fixed bin,
     ofile char (168) aligned,                           /* filename */
     st bit (72) aligned,                                /* ios status */
     statp ptr,                                          /* ptr to st */
     1 statx based (statp) aligned,                      /* status breakdown */
     2 fer fixed bin,
    (i, j) fixed bin,                                    /* temps */
     lth fixed bin;                                      /* bit length of input */

dcl 1 wordseg based (p1) aligned,                        /* scratch seg for words */
    2 ws (1000),
      3 wd char (32),                                    /* the word */
      3 wnx fixed bin,                                   /* fore ptr */
      3 ct fixed bin,                                    /* number of times seen */
      3 lp fixed bin,                                    /* index in lineseg for chain */
      3 zp fixed bin;                                    /* index of last entry of chain */

dcl 1 lineseg based (p2) aligned,                        /* scratch seg for crossreference */
    2 lst (1000),
```

```
      3 ls fixed bin,                                   /* index of start of line */
      3 ll fixed bin,                                   /* length of line */
      3 ln fixed bin,                                   /* chain to next entry */
      3 lo fixed bin;                                   /* line number in source */

dcl  welth fixed bin int static init (11);              /* 11 wds per wordseg entry */

dcl  buf (11) fixed bin;                                /* buffer for swap */

dcl (hcs_$initiate_count, hcs_$terminate_noname, hcs_$terminate_seg,
     hcs_$delentry_seg, hcs_$make_seg) ext entry,
    (ioa_, com_err_, expand_path_, cu_$arg_ptr, ios_$write_ptr,
     ioa_$ioa_stream, ios_$attach, ios_$detach,
     move_, parse_file_$parse_file_ptr,
     parse_file_$parse_file_line_no,
     parse_file_$parse_file_cur_line, parse_file_$parse_file_init,
     parse_file_$parse_file_init_ptr, parse_file_$parse_file_unset_break) ext entry;

/* - - - - - - - - - - - - - - - - - - - - - - */

/* Set up and get args. */

          unspec (NULC) = "001111111"b;                 /* set up idle char */
          call cu_$arg_ptr (1, ap, al, ec);             /* get name of file */
          if ec ^= 0 then go to er;
          path1 = substr (ap -> bchr, 1, al);           /* Pick up relative path. */
          out = substr (path1, 1, al) || ".kwoc";       /* generate filename */
          out1 = al + 5;


          call hcs_$make_seg ("", "", "", 1011b, p1, ec);   /* Create wordseg. */
          if ec ^= 0 then go to er;
          call hcs_$make_seg ("", "", "", 1011b, p2, ec);   /* Create lineseg. */
          if ec ^= 0 then go to er;

/* get list of words to suppress */

          call cu_$arg_ptr (2, ap, al, ec);             /* get name of suppress file */
          if ec ^= 0 | al = 0 then go to nosupr;        /* If none supplied, skip. */
          path = substr (bchr, 1, al);                  /* Get file name. */
          call parse_file_$parse_file_init (path, p, ec);   /* Set up to read file. */
          if ec ^= 0 then do;
er:            call com_err_ (ec, "kwoc", path1);
               return;
          end;
          call parse_file_$parse_file_unset_break (">_");

lp1:      call parse_file_$parse_file_ptr (q, cc, bk, ec);  /* Get one word to ignore. */
          if ec ^= 0 then go to endsupr;                /* Check for eof. */
          if bk ^= 0 then go to lp1;                    /* Ignore punctuation. */
          word = substr (q -> bchr, 1, cc);             /* pick off word */
          letter = substr (word, 1, 1);                 /* get first letter */
          i = index (alphabet, letter);                 /* find position in leadin vector */
          if i > 26 then i = i - 26;                    /* map caps to small */
          if v(i) = 0 then do;                          /* first time this letter used? */
               nw = nw + 1;                             /* yup, make a dummy */
               v(i) = nw;                               /* .. for the head of the list */
               wd(nw) = "";
               ct(nw) = -1;                             /* but don't index it */
```

```
          end;
          nw = nw + 1;                                  /* Allocate wordseq entry. */
          wd(nw) = word;                                /* fill in entry */
          ct(nw) = -1;                                  /* flag suppress */
          wnx(nw) = wnx(v(i));                          /* chain in entry */
          wnx(v(i)) = nw;                               /* ... after dummy */
          go to lp1;                                    /* Go get next word. */

endsupr:  call hcs_$terminate_seg (p, 0, ec);           /* Done with suppress file. */
          noo = nw;                                     /* count suppressed wds. */

/* now set up for reading file */

nosupr:   call parse_file_$parse_file_init (path1, pp, ec); /* Set up to read input. */
          if ec ^= 0 then go to er;
          call parse_file_$parse_file_unset_break (">_");

loop:     call parse_file_$parse_file_ptr (q, cc, bk, ec); /* Get one word from file. */
          if ec ^= 0 then go to endin;                  /* Check eof. */
          if bk ^= 0 then go to loop;                   /* Ignore punctuation. */
          word = substr (q -> bchr, 1, cc);             /* get one word of input */
          letter = substr (word, 1, 1);                 /* get first letter */
          j = index (alphabet, letter);                 /* get index in "v" */
          if j > 26 then j = j - 26;                    /* map caps to smalls */

          if v(j) = 0 then do;                          /* first time this letter? */
               nw = nw + 1;                             /* yes. make a dummy */
               v(j) = nw;                               /* for the head of the list */
               wd(nw) = "";                             /* with a blank word */
               ct(nw) = -1;                             /* and a suppress flag */
          end;

          i = v(j);                                     /* ptr to first wd with this letter */

sh:       if wd(i) = word then do;                      /* is current word same as this? */
               if ct(i) = -1 then go to loop;           /* Yes. Is indexing suppressed? */
found:         call parse_file_$parse_file_cur_line (ci, cc); /* find line */
               if ct(i) > 0 then do;                    /* check for dup */
                    if ci = ls(zp(i))+1 then if cc = ll(zp(i)) then go to loop;
               end;
               call parse_file_$parse_file_line_no (lno); /* get line number */
               ct(i) = ct(i) + 1;                       /* Count hits on oword. */
               nx = nx + 1;                             /* alloc crossreference */
               ls(nx) = ci-1;                           /* Fill in ptrs to line-start & line length */
               ll(nx) = cc;                             /* use cr on line */
               lo(nx) = lno;                            /* remember line no */
               if lp(i) = 0 then lp(i) = nx;            /* chain in */
               else ln(zp(i)) = nx;                     /* set chain ptr in last elem */
               zp(i) = nx;                              /* update wd ptr to last elem */

               go to loop;
          end;
          if wnx(i) = 0 then do;                        /* was this the last word in chain? */
               nw = nw + 1;                             /* yes, so add word to list */
               wcount(j) = wcount(j) + 1;
               wnx(i) = nw;                             /* chain word in */
               wd(nw) = word;                           /* fill in entry */
               i = nw;                                  /* make it look just like a regular hit */
               go to found;                             /* ... */
```

```
          end;
          i = wnx(i);                                   /* more words to search, up + at end */
          go to sh;                                     /* ... */

/* now sort wordseg */

endin:    call ioa_ ("^d references to ^d words in ""^a""", nx, nw-noo, path1);

sort:     do j = 0 to 26;                               /* sort each letter separately */
pass:          i = v(j);                                /* pick up index of first on list */
               if i = 0 then go to nosort;              /* if unused letter */
               i = wnx(i);                              /* now you see why the dummy entry! */
               swap = 0;                                /* clear swap switch */
cmp:           if wnx(i) = 0 then if swap = 0 then go to nosort;
               else go to pass;                         /* if at bottom, see if another pass */
               if wd(i) > wd(wnx(i)) then do;           /* compare to next on list */
                    word = wd(i);                       /* out of order - swap */
                    wd(i) = wd(wnx(i));                 /* ... */
                    wd(wnx(i)) = word;
                    ti = lp(i);
                    lp(i) = lp(wnx(i));
                    lp(wnx(i)) = ti;
                    ti = ct(i);
                    ct(i) = ct(wnx(i));
                    ct(wnx(i)) = ti;
                    swap = swap + 1;                    /* remember swap */
               end;
               i = wnx(i);                              /* advance i */
               go to cmp;
nosort:   end sort;
          call ioa_ ("lists sorted, begin output.");

/* now loop thru file & print index */

          call expand_path_ (addr (out), outl, addr (ofile), null, ec);
          call ios_$attach (out, "file_", ofile, "", st);   /* attach output stream */
          statp = addr (st);
          ec = fer;                                     /* check error */
          if ec ^= 0 then go to er;

          do j = 0 to 26;                               /* do all the letters */
               i = v(j);
               if i = 0 then go to end2;
               if wcount(j) = 0 then go to end2;        /* skip letters with no word */
               if j > 0 then letter = substr (alphabet, j, 1);
               else letter = " ";
               call ioa_$ioa_stream (out, ""|^-Index for ^a - ^d words beginning with ^a^/^/",
               path1, wcount(j), letter);
lp3:           if ct(i) = -1 then do;                   /* Skip suppressed word. */
                    go to end1;
               end;
               else call ioa_$ioa_stream (out,""^/""^a"" ^d", wd(i), ct(i));
               ti = lp(i);                              /* Get start of crossref chain. */

lp2:           buffer = substr (pchr, ls(ti)+1, ll(ti)-1);  /* get line */
               do oi = 1 to ll(ti)-1;
                    if substr (buffer, oi, 1) = "%" then    /* change percent to o177 */
                    substr (buffer, oi, 1) = NULC;
               end;
```

```
               call ioa_$ioa_stream (out,""5x^a"", buffsc);  /* write line out */
               ti = ln(ti);                                  /* follow chain */
               if ti ^= 0 then go to lp2;                    /* Zero next ptr flags end. */
end1:          if wnx(i) ^= 0 then do;
                    i = wnx(i);
                    go to lp3;
               end;
end2:     end;

          call ios_$detach (out, "", "", st);

          call ioa_ ("");                                    /* Whew. */
          call hcs_$terminate_seg (pp, 0, ec);               /* Terminate input. */
          call hcs_$delentry_seg (p1, ec);                   /* Delete scratch. */
          call hcs_$delentry_seg (p2, ec);                   /* .. */

     end kwoc;
```

```
remark: Multics (multiplex%ed information and computing service) is a time-sharing system on the GE645
remark: the Multics Programmer's Manual (MPM) is the reference manual for the system
remark: a user normally control%s his Multics process from a typewriter terminal (console)
remark: a "process" is a locus of control within an instruction sequence
remark: a process has a two-dimension%al address space consist%ing of segment%s
remark: a "segment" is an address%able collection of word%s which has name%s and access attribute%s
remark: segment%s are divided into 1024-word page%s which are brought into core by the file system on demand
remark: a "directory" is a system-maintained table; each entry contain%s the name%s and attribute%s of a branch or link
remark: a "branch" may be a file branch, for a segment, or a directory branch, for an inferior directory
remark: a link entry gives the path name of some other entry; it is like an indirect reference
remark: every known segment in a process is associate%d with some file or directory branch
remark: each process has a process directory, where per-process segment%s are kept
remark: a "path name" is a sequence of entry name%s separated by ">" which specifies an entry
remark: an absolute Path name begins with ">" and leads from the root directory to some entry
remark: a relative path name is relative to the current working directory; it begins with "<" or a name
remark: the KST (known segment table) is a per-process table of reference name%s and segment number%s
remark: when one program first call%s another, a "linkage fault" occurs and the "linker" is call%ed
remark: the linker search%es for the call%ed segment and entrypoint and sets a pointer in the linkage section
remark: if a linkage fault can%not be satisfied, the "linkage_error" condition is signal%led
remark: the segment%s address%able by a process are grouped into ring%s of protection
remark: a user can enter a lower-number%ed protection ring only through a "gate" segment
remark: the gate segment "hcs_" allow%s the user to call into the ring 0 (hardcore) supervisor
remark: an access control list (ACL) is a per-branch list of item%s giving access rights to the branch
remark: each item on an ACL gives an access name and the access mode permit%ted
remark: an access name is of the form "user%name.project%name.instance"; any part may be "*"
remark: a user's access mode to a segment may be any combination of "rewa" (read, execute, write, append), or null
remark: a common access control list (CACL) is a list of item%s giving default access for branch%es in a directory
remark: a directory may have a record quota, which limit%s the number of page%s which may be charged to it
remark: if the record quota of a directory is 0, its page%s are charged to its parent directory
remark: inter-segment reference%s are of the form "segment%name$entrypoint"; both part%s are symbolic
remark: at login, a user's working directory and home directory are ">%user_dir_dir%>%project%>%name"
remark: an "object segment" is an executable segment; it has program text and a linkage section (see MPM RefData 3.4)
remark: a "command" is a procedure which can be call%ed from the console; its argument%s must be character string%s
remark: an "ioname" is the name of an input-output stream or device channel
remark: user%s make call%s to the input-output switch "ios_" to read and write on ioname%s
remark: the GIOC (generalized input-output control%ler) control%s all peripheral%s and terminal%s on the system
remark: a DCW (data control word) request%s the GIOC to perform some action on a device channel
remark: the GIOC interface module (GIM) controls device channel%s by means of DCW list%s
remark: each process has a call-save-return stack segment for each ring (see MPM RefData 2.3)
remark: each process has a combine%d linkage segment for each ring; it contain%s linkage section%s for each procedure
remark: an event channel is used to receive inter-process message%s and wakeup%s
remark: an event-call channel has an associate%d procedure which is call%ed when a wakeup is sent to the channel
remark: an event-wait channel records event message%s and wakeup%s; user%s call "ioc_$block" to wait for them
remark: "signal_" signal%s the "unclaimed_signal" condition if it can%not find a handler for a condition
remark: "quit" is signal%led after a "quit signal" (ATTN on 2741, INTERRUPT on model 37 Teletype)
remark: the 128 ASCII character%s are code%d as 7-bit character%s in 9-bit byte%s (see MPM RefData 3.1)
remark: MPM RefData 1.3 gives all the system error code%s
remark: the Standard Service System (SSS) is a group of carefully checked and efficient command%s and subroutine%s
remark: SSS command%s are marked "SSScmd" in this index; consult the SSS section of the MPM
remark: SSS active function%s are marked "SSS_af" in this index; consult the SSS section of the MPM
remark: SSS subroutine%s are marked "SSSsub" in this index; consult the SSS section of the MPM
remark: the Author-Maintained Library (AML) is a group of command%s and subroutine%s that do not qualify for SSS
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPM
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPM
SSScmd: addname (an) - add additional entry name%s to file branch
AMLcmd: adjust - set bit count on file branch
AMLcmd: adjust$block - set bit count on file branch, start from current length
AMLcmd: adjust$block_test - check bit count on file branch, start from current length
AMLcmd: adjust$test - check bit count on file branch
AMLcmd: adjust$test_block - check bit count on file branch, start from current length
```

```
SSScmd: archive (ac) - combine several file%s into an archive file
SSScmd: archive_sort (as) - sort an archive file on component file name%s
AMLcmd: ascii_check - check all character%s of a file for valid ASCII
AMLcmd: ascii_check$archive - check all character%s of an archive file for valid ASCII
SSScmd: basic - compile and execute a program in the basic language
AMLcmd: bcpl - compile a program in the bcpl language
SSScmd: bindarchive (ba) - create a bound object segment from an archive of object segment%s
SSScmd: bsys - interactive editor and terminal interface for program%ming in the basic language
SSScmd: calc - desk calculator with memory for function%s and expression%s
SSScmd: changevdir (cwd) - change current working directory
AMLcmd: compare_ascii (cpa) - compare two character file%s
SSScmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"
AMLcmd: convert_360_fortran (c360f) - convert program%s in the fortran language from 360 format
SSScmd: copy (cp) - copy a file
SSScmd: create (cr) - create a file branch
SSScmd: createdir (cd) - create a directory branch
AMLcmd: createsymtab (cst) - create symbol table file%s for pl1-language program%s (used with "debug")
AMLcmd: ctss_aarchv - extract all component file%s from a CTSS ASCII archive segment
AMLcmd: ctss_aarchv$aa_conv - convert CTSS ASCII archive file into Multics format
AMLcmd: debug (db) - symbolic interactive debug%ging aid
SSScmd: decam - desk calculator with memory
SSScmd: delete (dl) - delete a file
SSScmd: deleteacl (da) - delete item from an ACL
SSScmd: deletecacl (dc) - delete item from a CACL
SSScmd: deletedir (dd) - delete a directory branch and all file%s contain%ed in it
SSScmd: deleteforce (df) - delete a file branch; set write access if necessary
SSScmd: deletename (dn) - delete entry name from a file branch
SSScmd: dprint (dp) - request delayed print%ing of a file on high-speed printer
SSScmd: dpunch (dpn) - request delayed punch%ing of a file on high-speed punch
SSScmd: edm - context editor for character file%s
SSScmd: endfile - close fortran file%s
AMLcmd: entry_usage - print out usage count%s for entrypoint%s
AMLcmd: entry_usage$clear - set entrypoint usage count%s to 0
AMLcmd: eplbsa - request delayed assembly of program%s in the eplbsa language
SSScmd: exec_com (ec) - execute a series of command line%s contain%ed in a file
AMLcmd: extract_archive - extract all component file%s from a CTSS archive file
SSScmd: file_output (fo) - attach "user_output" to a given file instead of the console
SSScmd: fortran - compile program%s in the fortran language
AMLcmd: fs_chname - add, delete, or change entry name%s on a file
SSScmd: get_com_line (gcl) - print out current expand%ed command line length
SSS_af: get_pathname (gpn) - return absolute path name for given reference name
SSScmd: getquota (gq) - print out record quota and current use for given directory
SSScmd: global (gb) - execute a command in every directory inferior to a given directory
SSScmd: help - print out system information file%s
SSScmd: hold - save machine conditions after error
SSScmd: how_many_users (hmu) - print out current number of user%s and maximum number
AMLcmd: ind - indent program%s in the pl1 language
SSScmd: initiate (in) - make a file known by a given reference name
SSScmd: iocall - call "ios_" with given argument%s
SSScmd: iomode - set character conversion mode for an ioname
SSScmd: line_length (ll) - set maximum line length for an ioname
SSScmd: link (lk) - create a link entry
SSScmd: list (ls) - print out access, length, and entry name%s for given file%s in a directory
SSScmd: listacl (la) - print out ACL for a file
SSScmd: listcacl (lc) - print out CACL for a directory
SSScmd: listnames (ln) - print out entry name%s for file%s in a directory
SSScmd: listotals (lt) - print out entry count%s and use for a directory
SSScmd: login - identify user by name and project to system control and create user process
SSScmd: logout - request process destruction and session termination
```

```
AMLcmd: lower_case - convert all character%s in a file to lower case
AMLcmd: mail (ml) - send a file to another user's mailbox
AMLcmd: mail$unlock - un%lock a lock%ed mailbox
SSScmd: moveb (mv) - move a file to another directory
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
SSScmd: new_proc - request user control to create a new user process
SSScmd: page_trace (pgt) - print out information about last 200 page fault%s
SSS_af: pd - return absolute path name of process directory
AMLcmd: pht - print horizontal tab
SSScmd: pl1 - compile program%s in the pl1 language
SSScmd: pl1d: request delayed compile of program%s in the pl1 language
AMLcmd: pnl - print newline
SSScmd: print (pr) - print a character file
SSScmd: print_attach_table (pat) - print ioname attach%ment%s for process
SSScmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
SSScmd: print_object_map (pom) - print object map for an object segment
SSScmd: printhomedir (phd) - print absolute path name of home directory
AMLcmd: printsymtab (pst) - print symbol table for a pl1 program
SSScmd: printwdir (pwd) - print absolute path name of current working directory
SSScmd: probe - interactive debug%ging aid
SSScmd: program_interrupt (pi) - signal the condition "program_interrupt"
AMLcmd: ps - print argument string
AMLcmd: qed - program%mable context editor for character file%s
SSScmd: ready (rdy) - print listener ready message
SSScmd: ready_off (rdf) - turn off listener ready message%s
SSScmd: ready_on (rdn) - turn on listener ready message%s
SSScmd: release - release machine conditions save%d by "hold"
SSScmd: rename (rn) - change an entry name for a file
SSScmd: reorder_archive - change order of component file%s in an archive file
AMLcmd: return_to - return to a given stack frame
AMLcmd: return_to$repeat_call - return to given stack frame and repeat call
AMLcmd: runoff - format a character file for print%ing
SSScmd: set_bit_count (sbc) - set bit count for a file branch
SSScmd: set_com_line (scl) - set maximum expand%ed command line length
SSScmd: setacl (sa) - add item to ACL for a file
SSScmd: setcacl (sc) - add item to CACL for a directory
SSScmd: sethomedir (shd) - set path name of home directory
SSScmd: start - restart execution at save%d machine conditions
SSScmd: status (st) - print out status information from directory entry for a branch
SSScmd: terminate (tm) - terminate a reference name
SSScmd: terminate_refname (tnr) - terminate a segment given by reference name
SSScmd: terminate_segno (tns) - terminate a segment given by segment number
SSScmd: unlink (ul) - delete a link entry from a directory
SSS_af: wd - return absolute path name of current working directory
SSScmd: who - print out name%s of user%s on system
SSScmd: who$admin - print out name%s, project id's, and process id's for user%s on system
SSScmd: who$long - print out name%s and project id's for user%s on system
SSSsub: acm_$sleep_int_secs - block process for time interval in second%s
SSSsub: acm_$sleep_interval - block process for time interval in microsecond%s
SSSsub: acm_$sleep_time - block process until time given in microsecond%s
SSSsub: acm_$wakeup_int_secs - request wakeup over event channel after interval in second%s
SSSsub: acm_$wakeup_interval - request wakeup over event channel after interval in microsecond%s
SSSsub: acm_$wakeup_time - request wakeup over event channel at time given in microsecond%s
SSSsub: active_fnc_err_ - write error message on "user_output" and signal condition "active_function_error"
SSSsub: alloc_ - allocate storage in given area, return pointer
SSSsub: area_ - initialize an area
SSSsub: area_$redef - change the length of an area
SSSsub: check_star_ - check whether path name contain%s a star
SSSsub: clock_ - return calendar clock time in microsecond%s
```

```
SSSsub: com_err_ - format system error message and write on "user_output"
SSSsub: com_err_$check_fs_errcode_ - return character string for system error code
SSSsub: com_err_$notify - format non-system error message and write on "user_output"
SSSsub: condition_ - establish handler for a condition
SSSsub: cu_$arg_count - return number of argument%s of call%ing procedure
SSSsub: cu_$arg_list_ptr - return pointer to call%ing procedure's argument list
SSSsub: cu_$arg_ptr - return pointer to character-string argument
SSSsub: cu_$cl - call current listener to re-enter command level
SSSsub: cu_$cp - call current command processor to execute command line
SSSsub: cu_$gen_call - call procedure given by pointer with given argument list
SSSsub: cu_$grow_stack_frame - increase size of call%ing procedure's stack frame
SSSsub: cu_$level_get - return current validation level
SSSsub: cu_$level_set - set validation level
SSSsub: cu_$ptr_call - call procedure given by pointer with given argument%s
SSSsub: cu_$set_cl - define current listener
SSSsub: cu_$set_cp - define current command processor
SSSsub: cu_$shrink_stack_frame - reduce size of call%ing procedure's stack frame
SSSsub: cu_$stack_frame_ptr - return pointer to stack frame of call%ing procedure
SSSsub: cu_$stack_frame_size - return size of call%ing procedure's stack frame
SSSsub: cv_bin_ - convert binary to character string, any base
SSSsub: cv_bin_$dec - convert binary to character string, base 10
SSSsub: cv_bin_$oct - convert binary to character string, base 8
SSSsub: cv_dec_ - convert character string to binary, base 10
SSSsub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSsub: cv_oct_ - convert character string to binary, base 8
SSSsub: cv_oct_check_ - convert character string to binary, base 8; return error code
SSSsub: date_time_ - convert calendar clock time to character string date and time
SSSsub: date_time_$fstime - convert file system time to character string date and time
SSSsub: decode_object_ - return pointer%s to text, linkage, and symbol section of object segment
SSSsub: equal_ - expand entry name which may contain equal sign%s
SSSsub: expand_path_ - expand relative path name into absolute path name
SSSsub: freen_ - free allocate%d storage in an area
SSSsub: get_at_entry_ - return outer module, attach%ed ioname, and mode for given ioname
SSSsub: get_group_id_ - return user's access control name
SSSsub: get_group_id_$tag_star - return user's access control name with instance tag a star
SSSsub: get_pdir_ - return absolute path name of process directory
SSSsub: get_process_id_ - return user process id
SSSsub: get_ring_ - return the current ring number
SSSsub: hcs_$acl_add - add or change item%s on the ACL or CACL of a branch
SSSsub: hcs_$acl_add1 - add or change single ACL or CACL item for a branch
SSSsub: hcs_$acl_delete - delete all or part of the ACL or CACL of a branch
SSSsub: hcs_$acl_list - return contents of the ACL or CACL of a branch
SSSsub: hcs_$acl_replace - replace the entire ACL or CACL of a branch
SSSsub: hcs_$append_branch - create a file branch entry in parent directory and add user to ACL
SSSsub: hcs_$append_branchx - create directory or file branch entry in parent directory, given ACL
SSSsub: hcs_$append_link - create a link entry in parent directory
SSSsub: hcs_$assign - request SIM to assign and initialize device channel
SSSsub: hcs_$assign_linkage - allocate storage in linkage section
AMLsub: hcs_$block - block process until next wakeup
SSSsub: hcs_$chname_file - add and delete entry name%s on branch or link entry
SSSsub: hcs_$chname_seg - add and delete entry name%s on branch or link entry
SSSsub: hcs_$crawl_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_$del_dir_tree - delete all branch%es inferior to a directory
SSSsub: hcs_$delentry_file - delete branch or link entry from parent directory
SSSsub: hcs_$delentry_seg - delete entry for a segment from parent directory
SSSsub: hcs_$fs_get_brackets - return ring brackets and access for a segment from KST
SSSsub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_$fs_get_mode - return access of user with respect to segment
SSSsub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment
```

SSSsub: hcs_$fs_get_ref_name - return reference name%s for a segment from KST
SSSsub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_$fs_move_file - move contents from one file branch to another
SSSsub: hcs_$fs_move_seg - move contents from one segment to another
SSSsub: hcs_$fs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_$fs_search_set_wdir - change current working directory
SSSsub: hcs_$get_cur_status - return GIM position in DCW list
SSSsub: hcs_$get_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_$get_page_trace - return information about last 200 page fault%s
SSSsub: hcs_$get_status - return GIM status for device channel
SSSsub: hcs_$initiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSsub: hcs_$list_change - request GIM to change DCW list
SSSsub: hcs_$list_connect - request GIM to start device channel on DCW list
SSSsub: hcs_$list_size - set size of GIM DCW list for device channel
SSSsub: hcs_$make_ptr - return pointer to entrypoint given segment name and entrypoint name (search%es and make%s known)
SSSsub: hcs_$make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSsub: hcs_$proc_info - return process id, process group id, process directory name, ring number
SSSsub: hcs_$quota_get - return record quota, current use, and cumulative use for directory branch
SSSsub: hcs_$safety - request GIM to stop device channel and safety DCW list
SSSsub: hcs_$set_alarm - request wakeup at given time in microsecond%s
SSSsub: hcs_$set_bc - set bit count in file branch entry
SSSsub: hcs_$set_consistsw - set consistency switch in branch entry
SSSsub: hcs_$set_copysw - set copy switch in branch entry
SSSsub: hcs_$set_relatesw - set relate switch in branch entry
SSSsub: hcs_$set_timer - request wakeup or fault after given CPU time interval
SSSsub: hcs_$star_ - search a directory for entry name%s match%ing template
SSSsub: hcs_$star_list_ - search a directory for entry name%s match%ing template, return long information
SSSsub: hcs_$status - return information from directory entry about a branch or link
SSSsub: hcs_$status_long - return all information from directory entry about a branch or link
SSSsub: hcs_$status_minf - return bit count and type for a branch or link
SSSsub: hcs_$status_mins - return bit count and type for a segment
SSSsub: hcs_$terminate_file - terminate all reference name%s for branch from KST, make segment un%known
SSSsub: hcs_$terminate_name - terminate one reference name for segment from KST
SSSsub: hcs_$terminate_noname - terminate a null reference name for a segment from KST
SSSsub: hcs_$terminate_seg - terminate all reference name%s for a segment from KST, make segment un%known
SSSsub: hcs_$truncate_file - truncate file branch from a given offset
SSSsub: hcs_$truncate_seg - truncate segment from a given offset
SSSsub: hcs_$unassign - request GIM to un-assign a device channel
SSSsub: hcs_$usage_values - return cumulative page wait%s and CPU time used
AMLsub: hcs_$wakeup - send inter-process wakeup to process with given id
SSSsub: ioa_ - format a message and write it on "user_output"
SSSsub: ioa_$ioa_stream - format a message and write it on given ioname
SSSsub: ioa_$ioa_stream_nnl - format a message and write it on given ioname, no newline
SSSsub: ioa_$nnl - format a message and write it on "user_output", no newline
SSSsub: ioa_$rs - format a message and return character string
SSSsub: ioa_$rsnnl - format a message and return character string, no newline
SSSsub: ios_$abort - abort outstanding transaction%s for an ioname
SSSsub: ios_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSsub: ios_$changemode - change mode of attach%ment of an ioname
SSSsub: ios_$detach - un-attach one ioname from another ioname
SSSsub: ios_$getdelim - return current read delimiters for ioname
SSSsub: ios_$getmode - return binary encoding of mode of attach%ment for ioname
SSSsub: ios_$getsize - return current element size for an ioname
SSSsub: ios_$iowait - wait for transaction%s on workspace asynchronous ioname
SSSsub: ios_$order - issue request to outer module associate%d with an ioname
SSSsub: ios_$read - read from an ioname
SSSsub: ios_$read_ptr - read from ioname "user_input"
SSSsub: ios_$readsync - set read synchronization mode for an ioname

```
SSSsub: ios_$resetread - delete unused read-ahead collected for an ioname
SSSsub: ios_$resetwrite - delete unused write-behind collected for an ioname
SSSsub: ios_$seek - set reference pointer%s associate%d with an ioname
SSSsub: ios_$setdelim - set current read delimiters for an ioname
SSSsub: ios_$setsize - set current element size for an ioname
SSSsub: ios_$tell - return value of reference pointer associate%d with an ioname
SSSsub: ios_$upstate - return status for asynchronous transaction on an ioname
SSSsub: ios_$worksync - set workspace synchronization mode for an ioname
SSSsub: ios_$write - write on an ioname
SSSsub: ios_$write_ptr - write on ioname "user output"
SSSsub: ios_$writesync - set write synchronization mode for an ioname
AMLsub: ipc_$block - block process until given event
AMLsub: ipc_$chn_1 - return name of this validation level's event channel 1
AMLsub: ipc_$chn_2 - return name of this validation level's event channel 2
AMLsub: ipc_$create_ev_chn - create an event channel
AMLsub: ipc_$cutoff - inhibit event channel for reading
AMLsub: ipc_$decl_ev_call_chn - make event channel into event-call type
AMLsub: ipc_$decl_ev_wait_chn - make event channel into event-wait type (default)
AMLsub: ipc_$delete_ev_chn - delete an event channel
AMLsub: ipc_$drain_chn - drain event channel of any pending wakeup
AMLsub: ipc_$mask_ev_calls - request ipc_$block not to interrogate event-call channel%s
AMLsub: ipc_$read_ev_chn - return event message from an event channel
AMLsub: ipc_$reconnect - enable event channel for reading (after cutoff)
AMLsub: ipc_$set_call_prior - request ipc_$block to interrogate event-call channel%s before event-wait channel%s
AMLsub: ipc_$set_wait_prior - request ipc_$block to interrogate event-wait channel%s before event-call channel%s
AMLsub: ipc_$unmask_ev_calls - request ipc_$block to interrogate event-call channel%s
SSSsub: make_obj_map_ - write an object map into object segment
SSSsub: move_ - move a block of word%s from one place to another (given by pointer%s)
SSSsub: parse_file_ - scan character file and return indices of next break or symbol
SSSsub: parse_file_$parse_file_cur_line - return indices of current line of character file
SSSsub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
SSSsub: parse_file_$parse_file_init_ptr - initialize parse_file_ with pointer to segment
SSSsub: parse_file_$parse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: parse_file_$parse_file_set_break - declare break character%s to parse_file
SSSsub: parse_file_$parse_file_unset_break - undeclare break character%s for parse_file_
AMLsub: random_$exponential - return random number from exponential distribution with mean 1
AMLsub: random_$exponential_seq - return array of random number%s from exponential distribution
AMLsub: random_$normal - return random number from normal distribution
AMLsub: random_$normal_ant - return random number from antithetic normal distribution
AMLsub: random_$normal_seq - return array of random number%s from normal distribution
AMLsub: random_$random_get_seed - return current seed for random number%s
AMLsub: random_$normal_ant_seq - return array of random number%s from antithetic normal distribution
AMLsub: random_$random_set_seed - set seed for random number%s
AMLsub: random_$uniform - return random number from uniform distribution
AMLsub: random_$uniform_ant - return random number from antithetic uniform distribution
AMLsub: random_$uniform_ant_seq - return array of random number%s from antithetic uniform distribution
AMLsub: random_$uniform_seq - return array of random number%s from uniform distribution
AMLsub: read_list_ - read in free-format variable%s from "user_input"
AMLsub: reverse_index - scan character string from right to left for given character
SSSsub: reversion_ - revert a condition handler
AMLsub: roundb - round a number to next multiple of 2**n
AMLsub: roundb$rndshftb - round a number to multiple of 2**n, divide by 2**n
AMLsub: roundb$rndshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
AMLsub: roundb$roundb36 - round a number to multiple of 36, round to multiple of 2**n
SSSsub: signal_ - signal the occurrence of a condition, call its handler
SSSsub: term_ - remove segment's linkage section, unsnap linkage pointer%s to segment, and terminate segment
SSSsub: term_$nomakeUnknown - same as term_ for segment given by pointer, but do not terminate
SSSsub: term_$refname - same as term_ for segment given by reference name
SSSsub: term_$seg_ptr - same as term_ for segment given by pointer
```

```
SSSsub: ti_$findata - turn scratch segment into data segment
SSSsub: ti_$finobj - turn scratch segment into object segment
SSSsub: ti_$getseg - set up a scratch segment
SSSsub: unique_bits_ - return unique bit string
SSSsub: unique_chars_ - convert bit string into unique character string
AMLsub: whoami_ - return user's name and project as given at login
AMLsub: write_list_ - write variable%s on "user_output"
```