

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.

1.3

2

3

4

) 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 endpoint usage counts to 0
SSSub: hcs_scravi_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 1.4 describes the login and logout commands and system messages
Remark: MPM RefData 1.5 describes the input-output system and device and output 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_schn_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

SSSub: cv_bin\$dec - convert binary to character string, base 10
SSSub: cv_dec_ - convert character string to binary, base 10
SSSub: 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_schn_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\$roundb36 - round a number to multiple of 36, round to multiple of 2**n

"200" 2

SSSCmd: page-trace (prt) - print out information about last 200 page faults
SSSub: hcs\$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\$rndshktb36 - 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

"360" 1

AMLCmd: convert_360_fortran (c360f) - 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 output 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"

Index for Multics - 56 words beginning with a)

"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 (ia) - 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_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 MPH
remark: AML subroutines are marked "AMSub" in this index; consult the AML section of the MPH

"AMLCmd" 32

remark: AML commands are marked "AMLCmd" in this index; consult the AML section of the MPH
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_checks\$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_archive - extract all component files from a CTSS ASCII archive segment
AMLCmd: ctss_archive_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: pht - 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

"AMSub" 38

remark: AML subroutines are marked "AMSub" in this index; consult the AML section of the MPH
AMSub: hcs_\$block - block process until next wakeup

AMLsub: hcs_swakeup - send inter-process wakeup to process with given id
 AMLsub: ipc_block - block process until given event
 AMLsub: ipc_schn_1 - return name of this validation level's event channel 1
 AMLsub: ipc_schn_2 - return name of this validation level's event channel 2
 AMLsub: ipc_create_ev_chn - create an event channel
 AMLsub: ipc_scutoff - inhibit event channel for reading
 AMLsub: ipc_sdecl_ev_call_chn - make event channel into event-call type
 AMLsub: ipc_sdelete_ev_val_chn - make event channel into event-val type (default)
 AMLsub: ipc_sdrain_chn - drain event channel of any pending wakeup
 AMLsub: ipc_smask_ev_calls - request ipc_block next to interrogate event-call channels
 AMLsub: ipc_sread_ev_chn - return event message from an event channel
 AMLsub: ipc_sreconnect - enable event channel for reading (after cutoff)
 AMLsub: ipc_sset_call_prior - request ipc_block to interrogate event-call channels before event-wait channels
 AMLsub: ipc_sset_val_prior - request ipc_block to interrogate event-wait channels before event-call channels
 AMLsub: ipc_sunmask_ev_calls - request ipc_block to interrogate event-call channels with mean 1
 AMLsub: random_sexponential - return random number from exponential distribution
 AMLsub: random_sunormal - return random number from normal distribution
 AMLsub: random_sunormal_ant - return random number from antithetic normal distribution
 AMLsub: random_sunormal_get_seed - return array of random numbers from normal distribution
 AMLsub: random_sunormal_ant_seed - return current seed for random numbers
 AMLsub: random_sunormal_ant_seq - return array of random numbers from antithetic normal distribution
 AMLsub: random_sunormal_set_seed - set seed for random numbers
 AMLsub: random_suniform - return random number from uniform distribution
 AMLsub: random_suniform_ant - return random number from antithetic uniform distribution
 AMLsub: random_suniform_ant_seq - return array of random numbers from antithetic uniform distribution
 AMLsub: random_suniform_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\$roundshftb - round a number to multiple of 2**n, divide by 2**n
 AMLsub: roundb\$roundshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
 AMLsub: roundb\$roundshftb36 - 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 MPH 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_aarchy - 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

SSS_af: pd - return absolute path name of process: iifactory
SSScmd: printhomedir (phd) - print absolute path name of home directory
SSScmd: printwdir (pwd) - print absolute path name of current working directory
SSS_af: wd - return absolute path name of current working directory
SSSsub: expand_path - expand relative path name into absolute path name
SSSsub: get_pdir - return absolute path name of process directory
SSSsub: hcs_ufs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_ufs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_ufs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_ufs_make_seg - create file branch given absolute path name and make known by given reference name in XST

"ac" 1

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

"access" 12

Remark: a "segment" is an addressable collection of words which has names and access attributes
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 "rwa" (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
SSScmd: deleteforce (if) - delete a file branch; set write access if necessary
SSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSSsub: get_group_id - return user's access control name
SSSsub: get_group_id_star - return user's access control name with instance; tag a star
SSSsub: hcs_ufs_get_brackets - return ring brackets and access for a segment from XST
SSSsub: hcs_ufs_get_mode - return access of user with respect to segment

"acl-add" 1

SSSsub: hcs_ufs_acl_add - add or change items on the ACL or CACL of a branch

"acl-add1" 1

SSSsub: hcs_ufs_acl_add1 - add or change single ACL or CACL item for a branch

"acl-delete" 1

SSSsub: hcs_ufs_acl_delete - delete all or part of the ACL or CACL of a branch

"acl-list" 1

SSSsub: hcs_ufs_acl_list - return contents of the ACL or CACL of a branch

"acl-replace" 1

SSSsub: hcs_ufs_acl_replace - replace the entire ACL or CACL of a branch

"acm" 6

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

"action" 1

Remark: a DCW (data control word) requests the GISC to perform some action on a device channel

"active" 1

Remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM

"active_inc_err" 1

SSSsub: active_fnc_err - write error message on "user_output" and signal condition "active_function_error"

```

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

"add" 9
SSCmd: addname (an) - add additional entry names to file branch
AMCmd: fs_chname - add, delete, or change entry names on a file
SSCmd: setacl (sa) - add item to ACL for a file
SSCmd: setacl (sc) - add item to CACL for a directory
SSSub: hcs_sacl_add - add or change items on the ACL or CACL of a branch
SSSub: hcs_sacl_add1 - add or change single ACL or CACL item for a branch
SSSub: hcs_sappend_branch - create a file branch entry in parent directory and add user to ACL
SSSub: hcs_schname_file - add and delete entry names on branch or link entry
SSSub: hcs_schname_ssg - add and delete entry names on branch or link entry

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

"addname" 1
SSCmd: 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
AMCmd: adjust - set bit count on file branch
AMCmd: adjust$block - set bit count on file branch, start from current length
AMCmd: adjust$block_test - check bit count on file branch, start from current length
AMCmd: adjust$test - check bit count on file branch
AMCmd: adjust$test_block - check bit count on file branch, start from current length

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

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

"aid" 2
AMCmd: debug (db) - symbolic interactive debugging aid
SSCmd: probe - interactive debugging aid

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

"allocate" 3
SSSub: alloc_ - allocate storage in given area, return pointer
SSSub: free_ - free allocated storage in an area
SSSub: hcs_sassign_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
AMCmd: mail (ml) - send a file to another user's mailbox
SSCmd: moveb (mv) - move a file to another directory
SSSub: hcs_$fs_move_file - move contents from one file branch to another

```



```

SSSSub: hcs_$fs_move_seq - move contents from one segment to another
SSSSub: los_$attach - associate one filename with another filename, a mode, and an outer module
SSSSub: los_$detach - un-attach one filename from another filename
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 "reva" (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 in component file names
  AMLCmd: ascii_check$archive - check all characters of an archive file for valid ASCII
  SSSCmd: bind$archive (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$aasconv - 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 in 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: free_ - free allocated storage in an area

"area_2" 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: llocal - call "ios_" with given arguments

```

```

AMLCmd: ps - print argument string
SSSSub: cu_sarg_count - return number of arguments of calling procedure
SSSSub: cu_sarg_list_ptr - return pointer to calling procedure's argument list
SSSSub: cu_sarg_ptr - return pointer to character string argument
SSSSub: cu_sgen_call - call procedure given by pointer with given arguments
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$normal_int_seq - return array of random numbers from antithetic normal distribution
AMLSub: random$uniform_int_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_$ioval - wait for transactions on workspace asynchronous ioname
SSSSub: ios_$upstate - return status for asynchronous transaction on an ioname

"attach" 6
SSSCmd: console_output (co) - attach "user_output" to the console ioname "user_1/3"
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_$attach - associate one ioname with another ioname, a mode, and an outer module
SSSSub: ios_$changeable - 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

```

Index for Multics - 19 words beginning with b:

"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 MPK 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

AM1cmd: 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
 AM1sub: hcs_\$block -- block process until given wakeup
 AM1sub: ipc_\$block -- block process until given event
 AM1sub: ipc_\$mask_ev_calls -- request ipc_\$block not to interrogate event-call channels before event-wait channels
 AM1sub: ipc_\$set_call_prior -- request ipc_\$block to interrogate event-wait channels before event-call channels
 AM1sub: ipc_\$set_wait_prior -- request ipc_\$block to interrogate event-wait channels before event-call channels
 AM1sub: 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

AM1cmd: 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
 AM1cmd: adjust\$block -- set bit count on file branch, start from current length.
 AM1cmd: adjust\$block_test - check bit count on file branch, start from current length
 AM1cmd: adjust\$test -- check bit count on file branch
 AM1cmd: 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_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_\$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_consistency -- set consistency switch in branch entry

SSSSub: hcs_\$set_copysv - set copy switch in branch entry
SSSSub: hcs_\$set_relativ - 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

"bays" 1

SSSCmd: bays - 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 MPH 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: deletecaci (dc) - delete item from a CACL
SSSCmd: listcaci (lc) - print out CACL for a directory
SSSCmd: setcaci (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_del - 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\$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 filenames
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: lcall - 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_\$grov_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: MPH RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

"card" 2

Remark: MPH RefData 1.6 describes punch-card input and control cards
Remark: MPH 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_del - add or change single ACL or CACL item for a branch
SSSSUB: hcs-\$fs_search_set_vdir - change current working directory
SSSSUB: hcs-\$list_change - request GIM to change DCW list
SSSSUB: ios-\$changeinfo - change node of attachment of an ioname

"changemode" 1

SSSSUB: ios-\$changemode - change mode of attachment of an ioname

"changepdir" 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 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: acm-\$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 unassign 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

AMSub: ipc_screate_ev_chn - create an event channel
 AMSub: ipc_scutoff - inhibit event channel for reading
 AMSub: ipc_sdecl_ev_all_chn - make event channel into event-call type
 AMSub: ipc_sdecl_ev_wait_chn - make event channel into event-wait type (default)
 AMSub: ipc_sdelete_ev_chn - delete an event channel
 AMSub: ipc_sdrain_chn - drain event channel of any pending wakeup
 AMSub: ipc_smask_ev_calls - request ipc_sblock not to interrogate event-call channels
 AMSub: ipc_sread_ev_chn - return event message from an event channel
 AMSub: ipc_sreconnect - enable event channel for reading (after cutoff)
 AMSub: ipc_sset_call_prio - request ipc_sblock to interrogate event-call channels before event-wait channels
 AMSub: ipc_sset_wait_prio - request ipc_sblock to interrogate event-wait channels before event-call channels
 AMSub: ipc_sunmask_ev_calls - request ipc_sblock to interrogate event-wait 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
 AMCmd: ascii_check - check all characters of a file for valid ASCII
 AMCmd: ascii_checksarchive - check all characters of an archive file for valid ASCII
 AMCmd: compare_ascii (cpa) - compare two character files
 SSScmd: edm - context editor for character files
 SSScmd: lmode - set character conversion mode for: an ioname
 AMCmd: lower_case - convert all characters in a file to lower case
 SSScmd: print (pr) - print a character file
 AMCmd: qed - programmable context editor for character files
 SSScmd: runoff - format a character file for printing
 SSSsub: com_err_\$hack_\$s_errcode - return character string for system error code
 SSSsub: cu_sarg_ptr - return pointer to character-string argument
 SSSsub: cv_bin - convert binary to character string, any base
 SSSsub: cv_bin_sdec - convert binary to character string, base 10
 SSSsub: cv_bin_soc - 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 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
 SSSsub: date_time - convert calendar clock time to character string date and time
 SSSsub: date_time_\$ftime - convert file system time to character string date and time
 SSSsub: lba_srs - format a message and return character string
 SSSsub: lba_srsnl - 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_sparse_file_cur_line - return indices of current line of character file
 SSSsub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol
 SSSsub: parse_file_sparse_file_set_break - declare break characters to parse_file
 SSSsub: parse_file_sparse_file_unset_break - undeclare break characters for parse_file
 AMSub: 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

AMCmd: adjust_block_test - check bit count on file branch, start from current length
 AMCmd: adjust_test - check bit count on file branch
 AMCmd: adjust_test_block - check bit count on file branch, start from current length
 AMCmd: ascii_check - check all characters of a file for valid ASCII
 AMCmd: ascii_checksarchive - check all characters of an archive file for valid ASCII
 SSSsub: check_star - check whether path name contains a star

SSSSub: com_err_\$chack_fs_errcode_ - return character string for system error code
SSSSub: com_err_\$notify - format non-system error message and write on "user_output"

"combination" 1

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

"combine" 2

remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
SSScmd: archive (ac) - combine several files into an archive file

"command" 12

remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
remark: MPM RefData 1.1 describes the login and logout commands and system messages
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: the Author-Maintained Library (AML) is a group of commands and subroutines that do not qualify for SSS
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: global (gb) - execute a command in every idirectory inferior to a given idirectory
SSScmd: set_com_line (scl) - set maximum expanded command line length
SSSSub: cu_scl - call current listener to re-enter command level
SSSSub: cu_scp - call current command processor to execute command line
SSSSub: cu_set_cp - define current command processor

"commands" 1

remark: AML commands are marked "AMLCmd" in this index; consult the AML section of the MPM

"common" 1

remark: a common access control list (CACL) is a list of items giving default access for branches in a directory

"compare" 1

AMLCmd: compare_ascii (cpa) - compare two character files

"compare_ascii" 1

AMLCmd: compare_ascii (cpa) - compare two character files

"compile" 5

SSScmd: basic - compile and execute a program in the basic language
AMLCmd: bcpl - compile a program in the bcpl language
SSScmd: fortran - compile programs in the fortran language
SSScmd: pl1 - compile programs in the pl1 language
SSScmd: pl1d: request delayed compile of programs in the pl1 language

"component" 4

SSScmd: archive_sort (as) - sort an archive file on component file names
AMLCmd: ctss_archive - extract all component files from a CTSS ASCII archive segment
AMLCmd: extract_archive - extract all component files from a CTSS archive file
SSScmd: reorder_archive - change order of component files in an archive file

"computing" 1

remark: Multics (multiplexed information and computing service) is a time-sharing system on the GC645

"condition" 8

remark: if a linkage fault cannot be satisfied, the "linkage_error" condition is signaled
remark: "signal_" signals the "undelained_signal" condition if it cannot find a handler for a condition
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: condition_ - establish handler for a condition

SSSSub: reversion_ - revert a condition handler
SSSSub: signal_ - signal the occurrence of a condition, call its handler

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

"conditions" 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-\$raw1_out_data - return machine conditions at last fault in ring 0

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

"consistency" 1
SSSSub: hcs-\$set_consistsw - set consistency switch in branch entry

"console" 4
remark: a user normally controls his Multics process from a typewriter terminal (console)
remark: a "command" is a procedure which can be called from the console; its arguments must be character strings
SSScmd: console_output (cb) - attach "user_output" to the console ioname "user_1/3"
SSScmd: file_output (fb) - attach "user_output" to a given file instead of the console

"console_output" 1
SSScmd: console_output (cb) - attach "user_output" to the console ioname "user_1/3"

"consult" 5
remark: SSS commands are marked "SSSCM1" in this index; consult the SSS section of the MPH
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPH
remark: SSS subroutines are marked "SSSSub" in this index; consult the SSS section of the MPH
remark: AML commands are marked "AMLcmd" in this index; consult the AML section of the MPH
remark: AML subroutines are marked "AMLsub" in this index; consult the AML section of the MPH

"contain" 6
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 combined linkage segment for each ring; it contains linkage sections for each procedure
SSScmd: deletedir (fd) - delete a directory branch and all files contained in it
SSScmd: exec_com (ec) - execute a series of command lines contained in a file
SSSSub: check_star - check whether path name contains a star
SSSSub: equal_ - expand entry name which may contain equal signs

"contents" 3
SSSSub: hcs-\$acl_list - return contents of the ACL or CACL of a branch
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

"context" 2
SSScmd: edm - context editor for character files
AMLcmd: qed - programmable context editor for character files

"control" 11
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: 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 (generalized input-output controller) controls all peripherals and terminals on the system
remark: a DW (data control word) requests the GIOC to perform some action on a device channel
remark: MPH RefData 1.6 describes punch-card input and control cards

```

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
SSSub: get_group_id - return user's access control name
SSSub: get_group_id_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: lmode - set character conversion mode for:an ioname

"convert" 13
AMLCmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
AMLCmd: ccss_aarchv$a_conv - convert CTSS ASCII archive file into Multics format
AMLCmd: lower_case - convert all characters in a file to lower case
SSSub: cv_bin_ - convert binary to character string, any base
SSSub: cv_bin_sdec - convert binary to character string, base 10
SSSub: cv_bin_soct - convert binary to character string, base 8
SSSub: cv_dec_ - convert character string to binary, base 10
SSSub: cv_dec_check_ - convert character string to binary, base 10; return error code
SSSub: cv_oct_ - convert character string to binary, base 8
SSSub: cv_oct_check_ - convert character string to binary, base 8; return error code
SSSub: date_time_ - convert calendar clock time to character string date and time
SSSub: date_time_sftime - convert file system time to character string date and time
SSSub: 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
SSSub: hcs_sset_copysv - 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 endpoints
AMLCmd: entry_usage$clear - set endpoint 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
SSSub: hcs_initialize_count - make a file known by reference name in KSR, return pointer and bit count
SSSub: hcs_sset_bc - set bit count in file branch entry
SSSub: hcs_sstatus_minf - return bit count and type for a branch or link
SSSub: hcs_sstatus_mins - return bit count and type for a segment

"cp" 2
SSScmd: copy (cp) - copy a file
SSSub: cu_scp - 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
SSSub: hcs_scravi_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 p11-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
SSSub: hcs_append_branch - create a file branch entry in parent directory and add user to ACL
SSSub: hcs_append_branchx - create a directory or file branch entry in parent directory, given ACL
SSSub: hcs_append_link - create a link entry in parent directory
SSSub: hcs_smoke_seg - create file branch given absolute path name and make known by given reference name in KST
AMLsub: ipc_screate_ev_chn - create an event channel

"create_ev_chn" 1
AMLsub: ipc_screate_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 p11-language programs. (used with "debug")

"dst" 1
AMLcmd: createsymtab (cst) - create symbol table files for p11-language programs. (used with "debug")

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

"cu" 15
SSSub: cu_sarg_count - return number of arguments of calling procedure
SSSub: cu_sarg_list_ptr - return pointer to calling procedure's argument list
SSSub: cu_sarg_ptr - return pointer to character string argument
SSSub: cu_sci - call current listener to re-enter command level
SSSub: cu_scp - call current command processor to execute command line
SSSub: cu_sgen_call - call procedure given by pointer with given argument list
SSSub: cu_sgrow_stack_frame - increase size of calling procedure's stack frame
SSSub: cu_slevel_get - return current validation level
SSSub: cu_slevel_set - set validation level
SSSub: cu_sptr_call - call procedure given by pointer with given arguments
SSSub: cu_sset_cl - define current listener
SSSub: cu_sset_cp - define current command processor
SSSub: cu_sshrink_stack_frame - reduce size of calling procedure's stack frame
SSSub: cu_sstack_frame_ptr - return pointer to stack frame of calling procedure
SSSub: cu_sstack_frame_size - return size of calling procedure's stack frame

"cumulative" 2
SSSub: hcs_quota_get - return Racore quota, current use, and cumulative use for directory branch
SSSub: hcs_usage_values - return cumulative page waits and CPU time used

```

```

"current" 25
Remarks: 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$stat_block - check bit count on file branch, start from current length
SSScmd: change$wdir (cwd) - change current working directory
SSScmd: get_com_line (gcl) - print out current expanded command line length
SSScmd: get_quota (qg) - print out record quota and current use for given directory
SSScmd: how_many_users (hnu) - print out current number of users and maximum number
SSScmd: print$wdir (pwi) - print absolute path name of current working directory
SSS-af: wd - return absolute path name of current working directory
SSSSub: cu_scl - call current listener to re-enter command level
SSSSub: cu_scp - call current command processor to execute command line
SSSSub: cu_slevel_get - return current validation level
SSSSub: cu_sset_cl - define current listener
SSSSub: cu_sset_cp - define current command processor
SSSSub: get_ring - return the current ring number:
SSSSub: hcs_sfs_search_get_wdir - Return absolute path name of current working directory
SSSSub: hcs_sfs_search_set_wdir - Change current working directory
SSSSub: hcs_squota_get - return record quota, current use, and cumulative use for directory branch
SSSSub: ios_sgetdelim - return current read delimiters for ioname
SSSSub: ios_sgetsize - return current element size for an ioname
SSSSub: ios_ssetdelim - set current read delimiters for an ioname
SSSSub: ios_ssetsize - set current element size for an ioname
SSSSub: parse_file_sparse_file_cur_line - return indices of current line of character file
AMLsub: random$frandom_get_seed - return current seed for random numbers

"cutoff" 2
AMLsub: ipc_scutoff - inhibit event channel for reading
AMLsub: ipc_sreconnect - enable event channel for reading (after cutoff)

"cv_bin" 3
SSSSub: cv_bin - convert binary to character string, any base
SSSSub: cv_bin_sdec - convert binary to character string, base 10
SSSSub: cv_bin_soct - 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: change$wdir (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
SSSub: hcs_sget_status - return SIM position in DCW list
SSSub: hcs_slist_change - request GIM to change DCW list
SSSub: hcs_slist_connect - request GIM to start device channel on DCW list
SSSub: hcs_slist_size - set size of SIM DCW list for device channel
SSSub: hcs_ssafty - 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
SSSub: ti_sfindata - turn scratch segment into data segment

"date" 2
SSSub: date_time_ - convert calendar clock time to character string date and time
SSSub: date_time_\$ftime - convert file system time to character string date and time

"date_time_" 2
SSSub: date_time_ - convert calendar clock time to character string date and time
SSSub: date_time_\$ftime - convert file system time to character string date and time

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

"dc" 1
SSScmd: deleteacl (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 p11-language programs (used with "debug")
AMLcmd: debug (db) - symbolic interactive debugging aid
SSScmd: probe - interactive debugging aid

"dec" 1
SSSub: cv_bin_sdec - convert binary to character string, base 10

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

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

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

"declare" 1
SSSub: parse_file_sparse_files_set_break - declare break characters to parse_file_

"decode_object_" 1
SSSub: 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
  AMISub: ipc_$decl_ev_valt_chn - make event channel into event-valt 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 (dph) - request delayed punching of a file on high-speed punch
  AMICmd: eplbsa - request delayed assembly of programs in the eplbsa language
  SSSCmd: plld: 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: deleteacl (ds) - 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
  AMICmd: 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
  AMISub: ipc_$delete_ev_chn - delete an event channel

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

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

"deleteacl" 1
  SSSCmd: deleteacl (ds) - 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
SSSub: ios-\$getdelim - return current read delimiters for lonaame
SSSub: ios-\$setdelim - set current read delimiters for an lonaame

"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 output 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 references
Remark: MPM RefData 3.5 describes magnetic tape formats
Remark: MPM RefData 3.7 describes punched card codes

"desk" 2
SSCmd: calc = desk calculator with memory for functions and expressions
SSCmd: decam = desk calculator with memory

"destruction" 1
SSCmd: logout = request process destruction and session termination

"detach" 1
SSSub: ios-\$detach = un-attach one lonaame from another lonaame

"device" 10
Remark: an "lonaame" 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 output module types
SSSub: hcs-\$assign - request GIM to assign and initialize device channel
SSSub: hcs-\$get_status - return GIM status for device channel
SSSub: hcs-\$list_connect - request GIM to start device channel on DCW list
SSSub: hcs-\$list_size - set size of GIM DCW list for device channel
SSSub: hcs-\$safety - request GIM to stop device channel and safety DCW list
SSSub: hcs-\$unassign - request GIM to unassign device channel

"df" 1
SSCmd: deleteforce (if) = 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 (CACLI) 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" and "projectname"
SSCmd: changeadir (cwd) = change current working directory

```

SSCCmd: createdir (cd) - create a directory branch;
SSCCmd: deletedir (fd) - delete a directory branch; and all files contained in it
SSCCmd: getquota (gq) - print out record quota and current use for given directory
SSCCmd: global (gb) - execute a command in every directory inferior to a given directory
SSCCmd: list (ls) - print out access, length, and entry names for given files in a directory
SSCCmd: listcacl (lc) - print out CACL for a directory
SSCCmd: listnames (ln) - print out entry names for files in a directory
SSCCmd: listtotals (lt) - print out entry counts and use for a directory
SSCCmd: move (mv) - move a file to another directory
SSCCmd: movequota (mq) - move record quota between parent directory and inferior directory
SSS_af: pd - return absolute path name of process directory
SSCCmd: printhomedir (ph) - print absolute path name of home directory
SSCCmd: printwdir (pw) - print absolute path name of current working directory
SSCCmd: setcacl (sc) - add item to CACL for a directory
SSCCmd: sethomedir (shd) - set path name of home directory
SSCCmd: status (st) - print out status information from directory entry for a branch
SSCCmd: unlink (ul) - delete a link entry from a directory
SSS_af: vd - return absolute path name of current working directory
SSSub: get_fdir - return absolute path name of process directory
SSSub: hcs-$append_branch - create a file branch entry in parent directory and add user to ACL
SSSub: hcs-$append_link - create a link entry in parent directory
SSSub: hcs-$del_dir_tree - delete all branches inferior to a directory
SSSub: hcs-$delentry_file - delete branch or link entry from parent directory
SSSub: hcs-$delentry_seg - delete entry for a segment from parent directory
SSSub: hcs-$fs_get_path_name - return absolute path name of parent directory for a segment
SSSub: hcs-$fs_get_vdir - return absolute path name of parent directory and entry name of a segment
SSSub: hcs-$fs_search_get_vdir - return absolute path name of current working directory
SSSub: hcs-$fs_search_set_vdir - change current working directory
SSSub: hcs-$procinfo - return process id, process group id, process directory name, ring number
SSSub: hcs-$quota_get - return record quota, current use, and cumulative use for directory branch
SSSub: hcs-$star - search a directory for entry names matching template
SSSub: hcs-$star_list - search a directory for entry names matching template; return long information
SSSub: hcs-$status - return information from directory entry about a branch or link
SSSub: hcs-$status_long - return all information from directory entry about a branch or link

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

"divide" 2
AMSsub: roundb$endshftb - round a number to multiple of 2**n, divide by 2**n
AMSsub: roundb$endshftb36 - 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
SSCCmd: 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
AMSub: ipc-$drain_chn: - drain event channel of any pending Wakeup
"drain_chn" 1
AMSub: ipc-$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 MPA 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
 AM1cmd: 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
 AM1sub: ipc_\$reconnect - enable event channel for reading (after cutoff)

"encoding" 1
 SSSsub: ios_\$getmode - return binary encoding of node 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: adname (an) - add additional entry names to file branch
 SSScmd: deletename (dn) - delete entry name from a file branch
 AM1cmd: 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: listtotals (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_$append_branch_ - create a file branch entry in parent directory and add user to ACL
SSSSub: hcs_$append_branchx - create a 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_$chname_ssg - add and delete entry names on branch or link entry
SSSSub: hcs_$deleentry_file - delete branch or link entry from parent directory
SSSSub: hcs_$deleentry_ssg - delete entry for a segment from parent directory
SSSSub: hcs_$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_relatasw - 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)

"epibsa" 1
AMLCmd: epibsa - request delayed assembly of programs in the epibsa 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_inc_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_swakeup_int_secs - request wakeup over event channel after interval in seconds
SSSSub: acm_swakeup_interval - request wakeup over event channel after interval in microseconds
SSSSub: acm_swakeup_time - request wakeup over event channel at time given in microseconds
AMLSub: ipc_block - block process until given event
AMLSub: ipc_schn_1 - return name of this validation level's event channel 1
AMLSub: ipc_schn_2 - return name of this validation level's event channel 2
AMLSub: ipc_create_ev_chn - create an event channel
AMLSub: ipc_scutoff - inhibit event channel for reading
AMLSub: ipc_sdecl_ev_call_chn - make event channel into event-call type
AMLSub: ipc_sdecl_ev_wait_chn - make event channel into event-wait type (default)
AMLSub: ipc_sdelete_ev_chn - delete an event channel
AMLSub: ipc_sdrain_chn - drain event channel of any pending wakeup
AMLSub: ipc_smask_ev_calls - request ipc_block next to interrogate event-call channels
AMLSub: ipc_sread_ev_chn - return event message from an event channel
AMLSub: ipc_sreconnect - enable event channel for reading (after cutoff)
AMLSub: ipc_sset_call_prior - request ipc_block to interrogate event-call channels before event-wait channels
AMLSub: ipc_sset_wait_prior - request ipc_block to interrogate event-wait channels before event-call channels
AMLSub: ipc_sunmask_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 XPM RefData 3.4)

"execute" 5

Remark: a user's access mode to a segment may be any combination of "reva" (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_scp - 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_sexponential - return random number from exponential distribution with mean 1
AMLSub: random_sexponential_seq - return array of random numbers from exponential distribution

"exponential_seq" 1

AMLSub: random_sexponential_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
SSSScmd: page_trace (pgt) - print out information about last 200 page faults
SSSSsub: hcs_scrawl_out_data - return machine conditions at last fault in ring 0
SSSSsub: hcs_sget_page_trace - return information about last 200 page faults
SSSSsub: hcs_sset_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
SSSScmd: 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_block - check bit count on file branch
SSSScmd: archive (ac) - combine several files into an archive file
SSSScmd: 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
SSSScmd: copy (cp) - copy a file
SSSScmd: 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\$a3_conv - convert CTSS ASCII archive file into Multics format
SSSScmd: delete (dl) - delete a file
SSSScmd: deletedir (dl) - delete a directory branch and all files contained in it
SSSScmd: deleteforce (df) - delete a file branch; set write access if necessary
SSSScmd: deletename (fn) - delete entry name from a file branch
SSSScmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSSScmd: dpunch (dpr) - request delayed punching of a file on high-speed punch
SSSScmd: edm - context editor for character files
SSSScmd: endfile - close fortran files
SSSScmd: 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
SSSScmd: 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
SSSScmd: help - print out system information files
SSSScmd: initiate (in) - make a file known by a given reference name
SSSScmd: list (ls) - print out access, length, and entry names for given files in a directory
SSSScmd: listacl (la) - print out ACL for a file
AMLCmd: 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
SSSScmd: move (mv) - move a file to another directory
SSSScmd: print (pr) - print a character file
AMLCmd: qed - programmable context editor for character files
SSSScmd: rename (rn) - change an entry name for a file
SSSScmd: reorder_archive - change order of component files in an archive file
AMLCmd: runoff - format a character file for printing
SSSScmd: set_bit_count (sbc) - set bit count for a file branch
SSSScmd: setacl (sa) - add item to ACL for a file
SSSSsub: date_time_\$ftime - convert file system time to character string date and time

SSSSub: hcs_sappend_branch - create a file branch entry in parent directory and add user to ACL
 SSSSub: hcs_sappend_branchX - create directory of file branch entry in parent directory, given ACL
 SSSSub: hcs_sfs_move_file - move contents from one file branch to another
 SSSSub: hcs_sinitiate - make a file known by reference name in KST, return pointer
 SSSSub: hcs_sinitiate_count - make a file known by reference name in KST, return pointer
 SSSSub: hcs_smake_seg - create file branch given absolute path name and make known by given reference name in KST
 SSSSub: hcs_sset_bc - set bit count in file branch
 SSSSub: hcs_struncate_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_sparse_file_cur_line - return indices of current line of character file
 SSSSub: parse_file_sparse_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_sfindata - turn scratch segment into data segment

"finobj" 1

SSSSub: ti_sfinobj - 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.p:objectname.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: cuss_aarhvsaa_cony - convert USS ASCII archive file into Multics format

AMLCmd: runoff - format a character file for printing

SSSSub: con_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: lga - format a message and write it on "user_output"

SSSSub: lga_sioa_stream - format a message and write it on given lghame

SSSSub: lga_sioa_stream_nhl - format a message and write it on given lghame, no newline

SSSSub: lga_srs - format a message and write it on "user_output", no newline

SSSSub: lga_srsnhl - format a message and return character string

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

```

SSSub: cu_$grov_stack_frame - increase size of calling procedure's stack frame
SSSub: cu_$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSub: cu_$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSub: cu_$stack_frame_size - return size of calling procedure's stack frame

"free" 2
SSSub: free_ - free allocated storage in an area:
AM1cmd: read_list_ - read in free-format variables from "user_input"

"free" 1
SSSub: free_ - free allocated storage in an area:

"fs_chname" 1
AM1cmd: fs_chname - add, delete, or change entry names on a file

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

"fs_get_dir_name" 1
SSSub: hcs_$fs_get_dir_name - return absolute path name of parent directory for a segment

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

"fs_get_path_name" 1
SSSub: hcs_$fs_get_path_name - return absolute path name of parent directory and entry name of a segment

"fs_get_ref_name" 1
SSSub: hcs_$fs_get_ref_name - return reference names for a segment from KST

"fs_get_seg_ptr" 1
SSSub: hcs_$fs_get_seg_ptr - return pointer to segment given reference name

"fs_move_file" 1
SSSub: hcs_$fs_move_file - move contents from one file branch to another

"fs_move_seg" 1
SSSub: hcs_$fs_move_seg - move contents from one segment to another

"fs_search_get_vdir" 1
SSSub: hcs_$fs_search_get_vdir - return absolute path name of current working directory

"fs_search_set_vdir" 1
SSSub: hcs_$fs_search_set_vdir - change current working directory

"fstime" 1
SSSub: date_time_$fstime - convert file system time to character string date and time

"function" 2
remark: SSS active functions are marked "SSS_af" in this index; consult the SSS section of the MPM
SSSCmd: calc - desk calculator with memory for functions and expressions

```

"G6645" 1
remark: Multics (multiplexed information and computing service) is a time-sharing system on the G6645

"GIM" 9
remark: the GIOC interface module (GIM) controls device channels by means of DCW lists
SSSSub: hcs_sassign - request GIM to assign and initialize device channel
SSSSub: hcs_sget_cur_status - return GIM position in DCW list
SSSSub: hcs_sget_status - return GIM status for device channel
SSSSub: hcs_slist_change - request GIM to change DCW list
SSSSub: hcs_slist_connect - request GIM to start device channel on DCW list
SSSSub: hcs_slist_size - set size of SIM DCW list for device channel
SSSSub: hcs_safety - request GIM to stop device channel and safety DCW list
SSSSub: hcs_sunassign - request GIM to unassign 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_sget_entry_name - return endpoint 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_sgen_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_sget_cur_status - return GIM position in DCW list

"get_entry_name" 1
SSSSub: hcs_sget_entry_name - return endpoint 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_sget_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_sget_status - return SIM status for device channel

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

 "getmode" 1
 SSSsub: ios_sgetmode - 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_sgetseg - set up a scratch segment

 "getsize" 1
 SSSsub: ios_sgetsize - 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: local " 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 (trf) - terminate a segment given by reference name
 SSScmd: terminate_segno (trsg) - 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_sgen_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_branch - create directory or file branch entry in parent directory, given ACL
 SSSsub: hcs_sfs_get_seg_ptr - return pointer to segment given reference name
 SSSsub: hcs_sget_entry_name - return entrypoint name given offset in gate segment
 SSSsub: hcs_smake_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
 SSSsub: hcs_smake_seg - create file branch given absolute path name and make known by given reference name in KST
 SSSsub: hcs_sset_alarm - request wakeup at given time in microseconds
 SSSsub: hcs_sset_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: loa_sioa_stream - format a message and write it on given ioname
 SSSsub: loa_sioa_stream_nrl - format a message and write it on given ioname, no newline

AMISub: 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
AMISub: 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
AMISub: 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 perbranch 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

SSSend: global (gb) - execute a command in every directory inferior to a given directory

"gpn" 1

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

"gq" 1

SSSend: 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
SSSub: condition_ " establish handler for a condition
SSSub: reversion_ " revert a condition handler
SSSub: 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
SSSub: hcs_\$acl_adj " adj or change items on the ACL or CACL of a branch
SSSub: hcs_\$acl_adj1 " adj or change single ACL or CACL item for a branch
SSSub: hcs_\$acl_delete " delete all or part of the ACL or CACL of a branch
SSSub: hcs_\$acl_list " return contents of the ACL or CACL of a branch
SSSub: hcs_\$acl_replace " replace the entire ACL or CACL of a branch
SSSub: hcs_\$append_branch " create a file branch entry in parent directory and add user to ACL
SSSub: hcs_\$append_branchx " create a file branch entry in parent directory, given ACL
SSSub: hcs_\$append_link " create a link entry in parent directory
SSSub: hcs_\$assign " request GIM to assign and initialize device channel
SSSub: hcs_\$assign_linkage " allocate storage in linkage section
AMSub: hcs_\$block " block process until next wakeup
SSSub: hcs_\$cname_file " adj and delete entry names on branch or link entry
SSSub: hcs_\$cname_seg " adj and delete entry names on branch or link entry
SSSub: hcs_\$crawl_out_data " return machine conditions at last fault in ring 0
SSSub: hcs_\$del_dir_tree " delete all branches inferior to a directory
SSSub: hcs_\$delentry_file " delete branch or link entry from parent directory
SSSub: hcs_\$delentry_seg " delete entry for a segment from parent directory
SSSub: hcs_\$fs_get_brackets " return ring brackets and access for a segment from KST
SSSub: hcs_\$fs_get_dir_name " return absolute path name of parent directory for a segment
SSSub: hcs_\$fs_get_mode " return access of user with respect to segment
SSSub: hcs_\$fs_get_path_name " return absolute path name of parent directory from KST
SSSub: hcs_\$fs_get_ref_name " return reference names for a segment from KST
SSSub: hcs_\$fs_get_seg_ptr " return pointer to segment given reference name
SSSub: hcs_\$fs_move_file " move contents from one file branch to another
SSSub: hcs_\$fs_move_seg " move contents from one segment to another
SSSub: hcs_\$fs_search_get_wdir " return absolute path name of current working directory
SSSub: hcs_\$fs_search_set_wdir " change current working directory
SSSub: hcs_\$get_cur_status " return GIM position in DCW list
SSSub: hcs_\$get_entry_name " return entrypoint name given offset in gate segment
SSSub: hcs_\$get_page_trace " return information about last 200 page faults
SSSub: hcs_\$get_status " return GIM status for device channel
SSSub: hcs_\$initiate " make a file known by reference name in KST, return pointer
SSSub: hcs_\$list_change " make a file known by reference name in KST, return pointer and bit count
SSSub: hcs_\$list_change " request GIM to change JMW list
SSSub: hcs_\$list_connect " request GIM to start device channel on DCW list
SSSub: hcs_\$list_size " set size of GIM DCW list for device channel
SSSub: hcs_\$make_ptr " return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSub: hcs_\$make_seg " create file branch given absolute path name and make known by given reference name in KST
SSSub: hcs_\$proc_info " return process id, process group id, process directory name, ring number
SSSub: hcs_\$quota_get " return racori quota, current use, and cumulative use for directory branch
SSSub: hcs_\$safety " request GIM to stop device channel and safety DCW list
SSSub: hcs_\$set_alarm " request wakeup at given time in microseconds
SSSub: hcs_\$set_bc " set bit count in file branch entry
SSSub: hcs_\$set_consistency " set consistency switch in branch entry

SSSSub: hcs_\$set_relatesv - 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_minf - return bit count and type for a segment
SSSSub: hcs_\$status_minf - 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_name - 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 (dph) - request delayed punching of a file on high-speed punch

"hmu" 1

SSSCmd: hov_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

"hov_many_users" 1

SSSCmd: hov_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: whoAdmin - print out names, project id's, and process id's for users on system
SSScmd: whoLong - print out names and project id's for users on system
SSSub: get_proc_info - return user process id
SSSub: hcs_sproc_info - return process id, process group id, process directory name, ring number
AMSub: hcs_swakeup - 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
SSSub: cu_sgrow_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 "SSSub" 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 "AMSub" in this index; consult the AML section of the MPM

"indices" 2
SSSub: parse_file - scan character file and return indices of next break or symbol
SSSub: parse_file_sparse_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 inferior to a given inferior
SSScmd: movequota (mq) - move record quota between parent directory and inferior directory
SSSub: hcs_sdel_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 (pt) - print out information about last 200 page faults
SSScmd: print_link_info (pl) - print linkage and endpoint information for an object segment
SSScmd: status (st) - print out status information from directory entry for a branch
SSSub: hcs_sget_page_trace - return information about last 200 page faults
SSSub: hcs_sstar_list - search a directory for entry names matching template, return long information
SSSub: hcs_sstatus - return information from directory entry about a branch or link
SSSub: hcs_sstatus_long - return all information from directory entry about a branch or link

"inhibit" 1
AMSub: ipc_scutoff - inhibit event channel for reading

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

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

"initiate_count" 1
SSSub: 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 with "ios_" to read and write on ionames
Remark: the GIOC (generalized input-output controller) controls all peripherals and terminals on the system
Remark: MPH RefData 1.4 describes the input-output system and device and output module types
Remark: MPH RefData 1.6 describes punch-card input and control cards

"instance" 2
Remark: an access name is of the form "username,objectname,instance"; any part may be ""
SSSub: get_group_id_\$tag_star - return user's access control name with instance tag a star

"instead" 1
SSCmd: 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: MPH RefData 3.4 describes the linkage section and inter-segment references
AMSub: hcs_\$wakep - send inter-process wakeup to process with given id

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

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

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

"interval" 5
SSSub: acm-\$sleep_int_secs - block process for time interval in seconds
SSSub: acm-\$sleep_interval - block process for time interval in microseconds
SSSub: acm-\$wakep_int_secs - request wakeup over event channel after interval in seconds
SSSub: acm-\$wakep_interval - request wakeup over event channel after interval in microseconds
SSSub: 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_sioa_stream - format a message and write it on given ioname
SSSSub: ioa_sioa_stream_nnl - format a message and write it on given ioname, no newline
SSSSub: ioa_snnl - format a message and write it on "user_output", no newline
SSSSub: ioa_srs - format a message and return character string
SSSSub: ioa_srsnnl - format a message and return character string, no newline

"ioa_stream" 1
SSSSub: ioa_sioa_stream - format a message and write it on given ioname

"ioa_stream_nnl" 1
SSSSub: ioa_sioa_stream_nnl - format a message and write it on given ioname, no newline

"local" 1
SSScmd: local - 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_sioa_stream - format a message and write it on given ioname, no newline
SSSSub: ioa_sioa_stream_nnl - format a message and write it on given ioname
SSSSub: ios_abort - abort outstanding transactions for an ioname
SSSSub: ios_sattach - associate one ioname with another ioname, a mode, and an outer module
SSSSub: ios_schangeio - change mode of attachment for an ioname
SSSSub: ios_sdetach - un-attach one ioname from another ioname
SSSSub: ios_sgetdelim - return current read delimiters for ioname
SSSSub: ios_sgetmode - return binary encoding of mode of attachment for ioname
SSSSub: ios_sgetsize - return current element size for an ioname
SSSSub: ios_sioawait - wait for transactions on workspace asynchronous ioname
SSSSub: ios_sorder - issue request to outer module, associated with an ioname
SSSSub: ios_sread - read from an ioname
SSSSub: ios_sread_ptr - read from ioname "user_input"
SSSSub: ios_sreadsync - set read synchronization mode for an ioname
SSSSub: ios_sresetread - delete unused readahead collected for an ioname
SSSSub: ios_sresetwrite - delete unused write-behind collected for an ioname
SSSSub: ios_sseek - set reference pointers associated with an ioname
SSSSub: ios_ssetdelim - set current read delimiters for an ioname
SSSSub: ios_ssetsize - set current element size for an ioname
SSSSub: ios_sstell - return value of reference pointer associated with an ioname
SSSSub: ios_supdate - return status for asynchronous transaction on an ioname
SSSSub: ios_sworksync - set workspace synchronization mode for an ioname
SSSSub: ios_swrite - write on an ioname
SSSSub: ios_swrite_ptr - write on ioname "user_output"
SSSSub: ios_swritesync - 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: local - call "ios_" with given arguments
SSSSub: ios_abort - abort outstanding transactions for an ioname
SSSSub: ios_sattach - associate one ioname with another ioname, a mode, and an outer module
SSSSub: ios_schangeio - 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_\$resetraai - 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_\$setll - return value of reference pointer associated with an ioname
 SSSSub: ios_\$substate - 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_\$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-wait channels before event-call 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: deletetac1 (ia) - delete item from an ACL
 SSSCmd: deletetac1 (ic) - delete item from a CACL
 SSSCmd: setac1 (sa) - add item to ACL for a file
 SSSCmd: setac1 (sc) - add item to CACL for a directory
 SSSSub: hcs_\$acl_add - add or change items on the ACL of CACL of a branch
 SSSSub: hcs_\$acl_addf - 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 a segment from KST
SSSSub: hcs_\$terminate_name - 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 endpoint given segment name and endpoint 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 I.

"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 p11-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 p11 language
SSScmd: p11 - compile programs in the p11 language
SSScmd: p11d - request delayed compile of programs in the p11 language

"last" 3
SSScmd: page_trace (prt) - print out information about last 200 page faults
SSSub: hcs_scravl_out_data - return machine conditions at last fault in ring 0
SSSub: hcs_sget_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 iohame
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
SSSub: area_sizeof - 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
SSSub: cu_\$cl - call current listener to re-enter command level
SSSub: cu_\$level_get - return current validation level
SSSub: cu_\$level_set - set validation level
AMlsub: ipc_schn_1 - return name of this validation level's event channel 1
AMlsub: ipc_schn_2 - return name of this validation level's event channel 2

"level_get" 1
SSSub: cu_\$level_get - return current validation level

"level_set" 1

SSSub: cu_level_set - set validation level

"link" 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

SSCmd: exec_com (ec) - execute a series of command lines contained in a file
SSCmd: get_com_line (gcl) - print out current expanded command line length
SSCmd: line_length (ll) - set maximum line length for an ioname
SSCmd: set_com_line (scl) - set maximum expanded command line length
SSSub: cu_fcp - call current command processor to execute command line
SSSub: parse_file_sparse_file_cur_line - return indices of current line of character file.

"line_length" 1

SSCmd: 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
SSCmd: link (lk) - create a link entry
SSCmd: unlink (ul) - delete a link entry
SSSub: hcs_append_link - create a link entry in parent directory
SSSub: hcs_schname_file - add and delete entry names on branch or link entry
SSSub: hcs_schname_seg - add and delete entry names on branch or link entry
SSSub: hcs_sdelete_file - delete branch or link entry from parent directory
SSSub: hcs_status - return information from directory entry about a branch or link
SSSub: hcs_status_long - return all information from directory entry about a branch or link
SSSub: 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 RPM RefData 3.4)
Remark: each process has a combined linkage segment for each ring; it contains linkage sections for each procedure
Remark: RPM RefData 3.4 describes the linkage section and inter-segment references
SSCmd: print_link_info (pli) - print linkage and entrypoint information for an object segment
SSSub: decode_object - return pointers to text, linkage, and symbol section of object segment
SSSub: hcs_assign_linkage - allocate storage in linkage section
SSSub: 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: RPM 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: RPM RefData 2.2.1 lists all condition names which the system may signal
Remark: RPM RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
SSCmd: list (ls) - print out access, length, and entry names for given files in a directory

SSSSub: cu_sarg_list_ptr - return pointer to calling procedure's argument list
SSSSub: cu_sgen_call - call procedure given by pointer with given argument list
SSSSub: hcs_sget_cur_status - return SIM position in DCW list
SSSSub: hcs_slist_change - request SIM to change DCW list
SSSSub: hcs_slist_connect - request SIM to start device channel on DCW list
SSSSub: hcs_slist_size - set size of SIM DCW list for device channel
SSSSub: hcs_safety - request SIM to stop device channel and safety DCW list

"list_change" 1
SSSSub: hcs_slist_change - request SIM to change DCW list

"list_connect" 1
SSSSub: hcs_slist_connect - request SIM to start device channel on DCW list

"list_size" 1
SSSSub: hcs_slist_size - set size of SIM DCW list for device channel

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

"listcac1" 1
SSScmd: listcac1 (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_scl - call current listener to re-enter command level
SSSSub: cu_sset_cl - define current listener

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

"listtotals" 1
SSScmd: listtotals (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: MPH RefData 2.1 describes local conventions for Multics users

"lock" 1
AMCcmd: mailunlock - 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: MPH 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
AMISub: 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
SSSub: 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 "data" 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 output 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 "SSSCM" 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: cts_sarchvsa_cony - convert CTS 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_scravl_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

"make" 10
 remark: users make calls to the input-output switch "ios_" to read and write on ionames
 SSScmd: initiate (in) - make a file known by a given reference name
 SSSsub: hcs_initiate - make a file known by reference name in KSI, return pointer
 SSSsub: hcs_initiate_count - make a file known by reference name in KSI, return pointer and bit count
 SSSsub: hcs_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
 SSSsub: hcs_make_ptr - return pointer to entrypoint given segment name and make known by given reference name in KSI
 SSSsub: hcs_make_seg - create file branch given absolute path name and make known from KSI, make segment unknown
 SSSsub: hcs_terminate_file - terminate all reference names for a segment from KSI, make segment unknown
 SSSsub: hcs_terminate_seg - terminate all reference names for a segment from KSI, make segment unknown
 AMLsub: ipc_sdecl_ev_call_chh - make event channel into event-call type
 AMLsub: ipc_sdecl_ev_wait_chh - make event channel into event-wait type (default)

"make_obj_map" 1
 SSSsub: make_obj_map - write an object map into object segment

"make_ptr" 1
 SSSsub: hcs_make_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)

"make_seg" 1
 SSSsub: hcs_make_seg - create file branch given absolute path name and make known by given reference name in KSI

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

"map" 2
 SSScmd: print_obj_map (ppm) - print object map for an object segment
 SSSsub: make_obj_map - write an object map into object segment

"marked" 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

"mask_ev_calls" 1
 AMLsub: ipc_mask_ev_calls - request ipc_block not to interrogate event-call channels

"match" 2
 SSSsub: hcs_sstar - search a directory for entry names matching template
 SSSsub: hcs_sstar_list - search a directory for entry names matching template, return long information

"maximum" 3
 SSScmd: hov_many_users (hmd) - print out current number of users and maximum number
 SSScmd: line_length (ll) - set maximum line length for an ioname
 SSScmd: set_com_line (scl) - set maximum expanded command line length

"mean" 1
 AMLsub: random_exponential - return random number from exponential distribution with mean 1

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

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

"message" 16
 remark: an event channel is used to receive inter-process messages and wakeups

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_inc_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: loa_ - format a message and write it on "user_output"
SSSSub: loa_sioa_stream_ - format a message and write it on given ioname
SSSSub: loa_sioa_stream_nhl - format a message and write it on given ioname, no newline
SSSSub: loa_snl_ - format a message and write it on "user_output", no newline
SSSSub: loa_srs_ - format a message and return character string
SSSSub: loa_srsnhl - format a message and return character string, no newline
AMISub: ipc_stream_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

"mi" 1

AMCcmd: 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 "read", "write", "execute", "append", or "null"
SSScmd: lmode - 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_changenode - change mode of attachment of an ioname
SSSSub: ios_getmode - return binary encoding of mode for an 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 (GIV) 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_move_file - move contents from one file branch to another
SSSSub: hcs_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$roundshftb - round a number to multiple of 2**n, divide by 2**n
AMLsub: roundb$roundshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
AMLsub: roundb$roundshftb36 - round a number to multiple of 36, round to multiple of 2**n

"multiplex" 1
remark: Multics (multiplied information and computing service) is a time-sharing system on the GP645

"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.pobjectname.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: delete (dn) - delete entry name from a file branch
 SSScmd: fs_chname - add, delete, or change entry names on a file
 SSS_af: get_pathname (gn) - return absolute path name for a 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: rehomedir (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\$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_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_\$fs_get_dir_name - 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_\$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
 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, riny number
 SSSsub: hcs_\$star - search a directory for entry names matching template, return long information
 SSSsub: hcs_\$star_list - search a directory for entry names matching template
 SSSsub: hcs_\$terminate_file - terminate all reference names for branch from KST, make segment unknown

```

SSSub: hcs-$terminate_name - terminate one reference name for segment from KSI
SSSub: hcs-$terminate_name - terminate a null reference name for a segment from KSI
SSSub: hcs-$terminate_seq - terminate all reference names for a segment from KSI. Make segment unknown
AMSub: ipc-$chn_1 - return name of this validation level's event Channel 1
AMSub: ipc-$chn_2 - return name of this validation level's event Channel 2
SSSub: parse_file_$parse_file_init - initialize parse_file_ with given segment name
SSSub: term-$name - same as term_ for segment given by reference name
AMSub: whoami_ - return user's name and project as given at login

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

"nev" 1
SSCmd: nev_proc - request user control to create a new user process

"nev_proc" 1
SSCmd: nev_proc - request user control to create a new user process

"newline" 4
AMCmd: pnl - print newline
SSSub: ioa_$ioa_stream_nnl - format a message and write it on given iofame, no newline
SSSub: ioa_$nl - format a message and write it on "user_output", no newline
SSSub: ioa_$rannl - format a message and return character string, no newline

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

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

"no" 3
SSSub: ioa_$ioa_stream_nnl - format a message and write it on given iofame, no newline
SSSub: ioa-$nl - format a message and write it on "user_output", no newline
SSSub: ioa-$rannl - format a message and return character string, no newline

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

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

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

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

"normal_seq" 1
AMSub: 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

SSSSsub: 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 "reva" (read, execute, write, append), or null
SSSSsub: hcs_terminate_name - terminate a null reference name for a segment from KSI

"number" 24

remark: the KSI (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: hov_many_users (hnu) - print out current number of users and maximum number
SSScmd: terminate_segno (tms) - terminate a segment given by segment number
SSSSsub: cu_sarg_count - return number of arguments of calling procedure
SSSSsub: get_ring - return the current ring number:
SSSSsub: hcs_proc_info - return process 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_snormal - return random number from normal distribution
AMLSub: random_snormal_ant - return random number from antithetic normal distribution
AMLSub: random_snormal_seq - return array of random numbers from normal distribution
AMLSub: random_sranom_get_seed - return current seed for random numbers
AMLSub: random_snormal_ant_seq - return array of random numbers from antithetic normal distribution
AMLSub: random_suniform - return random number from uniform distribution
AMLSub: random_suniform_ant - return random number from antithetic uniform distribution
AMLSub: random_suniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLSub: random_suniform_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

Index for Multics. 13 words beginning with o

- "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_sfinobj -- 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_Soct -- convert binary to character string, base 8

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

- "offset" 3
 - SSSSub: hcs_sget_entry_name -- return entrypoint name given offset in gate segment
 - SSSSub: hcs_struncate_file -- truncate file branch from a given offset
 - SSSSub: hcs_struncate_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

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

- "outstanding" 1
 - SSSSub: ios_abort -- abort outstanding transactions for an ioname

Index for Multics - 55 words beginning with p)

"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 (pvt) - 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 faults and CPU time used

"page_trace" 1
SSSCmd: page_trace (pvt) - 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_branch - create a file branch entry in parent directory and add user to 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_sparse_file_cur_line - return indices of current line of character file
SSSSub: parse_file_sparse_file_init - initialize parse_file with given segment name
SSSSub: parse_file_sparse_file_init_ptr - initialize parse_file with pointer to segment
SSSSub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol
SSSSub: parse_file_sparse_file_set_break - declare break characters to parse_file
SSSSub: parse_file_sparse_file_unset_break - undeclare break characters for parse_file

"parse_file_cur_line" 1
SSSSub: parse_file_sparse_file_cur_line - return indices of current line of character file

"parse_file_init" 1
SSSSub: parse_file_sparse_file_init - initialize parse_file with given segment name

"parse_file_init_ptr" 1
SSSSub: parse_file_sparse_file_init_ptr - initialize parse_file with pointer to segment

"parse_file_ptr" 1
SSSSub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol

"parse_file_set_break" 1
SSSSub: parse_file_sparse_file_set_break - declare break characters to parse_file

"parse_file_unset_break" 1
SSSSub: parse_file_sparse_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 lonaame 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
SSSub: get_pathname (gpn) - return absolute path name for given reference name
SSSub: 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
SSSub: sethomedir (shd) - set path name of home directory
SSSub: wd - return absolute path name of current working directory
SSSub: check_star - check whether path name contains a star
SSSub: expand_path - expand relative path name into absolute path name
SSSub: get_pdir - return absolute path name of process directory
SSSub: hcs_sfs_get_dir_name - return absolute path name of parent directory for a segment
SSSub: hcs_sfs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSub: hcs_sfs_search_get_wdir - return absolute path name of current working directory
SSSub: hcs_sfs_make_seg - create file branch given absolute path name and make known by given reference name in KST

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

"pending" 1
AMSub: 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 (prt) - print out information about last 200 page faults

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

"pht" 1
AMCcmd: pht - print horizontal tab

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

"p11" 5
AMCcmd: createsymtab (cst) - create symbol table files for p11-language programs (used with "debug")
AMCcmd: ind - indent programs in the p11 language
SSScmd: p11 - compile programs in the p11 language

```

SSScmd: plid: request delayed compile of programs in the p11 language
AMLCmd: printsymtab (pnt) - print symbol table for a p11 program

"plid" 1

SSScmd: plid: request delayed compile of programs in the p11 language

"place" 1

SSSub: move - move a block of words from one place to another (given by pointers)

"pl1" 1

SSScmd: print_link_info (pl1) - 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
SSSub: alloc - allocate storage in given area, return pointer
SSSub: cu_sarg_list_ptr - return pointer to calling procedure's argument list
SSSub: cu_sarg_ptr - return pointer to character-string argument
SSSub: cu_sgencall - call procedure given by pointer with given argument list
SSSub: cu_sptr_call - call procedure given by pointer with given arguments
SSSub: cu_sstack_frame_ptr - return pointer to stack frame of calling procedure
SSSub: decode_object - return pointers to text, linkage, and symbol section of object segment
SSSub: hcs_sfs_get_seg_ptr - return pointer to segment given reference name
SSSub: hcs_sinitlate - make a file known by reference name in KSI, return pointer
SSSub: hcs_sinitiate_count - make a file known by reference name in KSI, return pointer and bit count
SSSub: hcs_smake_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSub: ios_stell - set reference pointers associated with an lname
SSSub: ios_stell - return value of reference pointer associated with an lname
SSSub: move - move a block of words from one place to another (given by pointers)
SSSub: parse_file_sparse_file_init_ptr - initialize parse_file with pointer to segment
SSSub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol
SSSub: term - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
SSSub: term_smakeunknown - same as term for segment given by pointer, but do not terminate
SSSub: term_sseg_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

SSSub: hcs_sget_cur_status - return 31M position in PCW 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_con_line (gcl) - print out current expanded command line length
SSScmd: get_quota (gq) - print out record quota and current use for given directory
SSScmd: help - print out system information files
SSScmd: how_many_users (hau) - 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: listacl (lc) - print out CACL for a directory
SSScmd: listnames (ln) - print out entry names for files in a directory
SSScmd: listtotals (lt) - print out entry counts and use for a directory
SSScmd: page_trace (prt) - print out information about last 200 page faults

```

AMLCmd: pht - print horizontal tab
AMLCmd: pnl - print newline
SSScmd: print (pf) - 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 (pdm) - print object map for an object segment
AMLCmd: printhomedir (phd) - print absolute path name of home directory
SSScmd: printsymtab (ps) - print symbol table for a pl1 program
AMLCmd: printvdir (pvd) - 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 (pdm) - 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 (ps) - print symbol table for a pl1 program

"printvdir" 1
SSScmd: printvdir (pvd) - print absolute path name of current working directory

"probe" 1
SSScmd: probe - interactive debugging aid

"proc_info" 1
SSScmd: 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
SSScmd: cu_sarg_count - return number of arguments of calling procedure
SSScmd: cu_sgen_list_ptr - return pointer to calling procedure's argument list
SSScmd: cu_sgenov_stack_frame - increase size of calling procedure's stack frame
SSScmd: cu_sptr_call - call procedure given by pointer with given arguments
SSScmd: cu_sshrink_stack_frame - reduce size of calling procedure's stack frame
SSScmd: cu_sstack_frame_ptr - return pointer to stack frame of calling procedure
SSScmd: cu_sstack_frame_size - return size of calling procedure's stack frame

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

```

remark: a "process" is a locus of control within an instruction sequence
 remark: every known segment in a process has a two-dimensional address space consisting of segments
 remark: each process 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
 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: 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 iohname 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
 AMIsub: hcs-\$block - block return process until next wakeup
 SSSsub: hcs-\$proc_info - return process id, process group id, process directory name, ring number
 AMIsub: hcs-\$wakeup - send inter-process wakeup to process with given id
 AMIsub: 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
 AMIcmd: bcpl - compile a program in the bcpl language
 SSScmd: bsyl - interactive editor and terminal interface for programming in the basic language
 AMIcmd: convert_360_fortran (c360f) - convert programs in the fortran language from 360 format
 AMIcmd: createsymbtab (cst) - create symbol table files for pl1-language programs. (used with "debug")
 AMIcmd: eplbsa - request delayed assembly of programs in the eplbsa language
 SSScmd: fortran - compile programs in the fortran language
 AMIcmd: 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
 AMIcmd: printsymtab (pst) - print symbol table for a pl1 program
 AMIcmd: 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
 AMIsub: whoami_ - return user's name and project is 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 (pwt) * print symbol table for: a pl1 program
"ptr_call" 1 SSSsub: cu_sptr_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 (dpm) * request delayed punching of: a file on high-speed punch
"pvd" 1 SSScmd: printvdir (pv3) * 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 (q) - print out record quota and current use for given directory
SSScmd: movequota (mq) - move record quota between Parent directory and inferior directory
SSSub: hcs_\$quota_get - return record quota, current use, and cumulative use for directory branch

"quota_get" 1
SSSub: hcs_\$quota_get - return record quota, current use, and cumulative use for directory branch

Index for Multics: 60 words beginning with r:

"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 1.4 describes the login and logout commands and system messages
remark: MPM RefData 1.6 describes the input-output system and device and output 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_Sexponential - return random number; from exponential distribution with mean 1
AMLSub: random_Sexponential_seq - return array of random numbers from exponential distribution
AMLSub: random_Snormal - return random number from normal distribution
AMLSub: random_Snormal_ant - return random number from antithetic normal distribution
AMLSub: random_Snormal_seq - return array of random numbers from normal distribution
AMLSub: random_Snormal_set_seed - return current seed for random numbers
AMLSub: random_Snormal_ant_seq - return array of random numbers from antithetic normal distribution
AMLSub: random_Srandom_set_seed - set seed for random numbers
AMLSub: random_Suniform - return random number from uniform distribution
AMLSub: random_Suniform_ant - return random number; from antithetic uniform distribution
AMLSub: random_Suniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLSub: random_Suniform_seq - return array of random numbers from uniform distribution

"random" 12

AMLSub: random_Sexponential - return random number; from exponential distribution with mean 1
AMLSub: random_Sexponential_seq - return array of random numbers from exponential distribution
AMLSub: random_Snormal - return random number from normal distribution
AMLSub: random_Snormal_ant - return random number from antithetic normal distribution
AMLSub: random_Snormal_seq - return array of random numbers from normal distribution
AMLSub: random_Snormal_set_seed - return current seed for random numbers
AMLSub: random_Snormal_ant_seq - return array of random numbers from antithetic normal distribution
AMLSub: random_Srandom_set_seed - set seed for random numbers
AMLSub: random_Suniform - return random number from uniform distribution
AMLSub: random_Suniform_ant - return random number; from antithetic uniform distribution
AMLSub: random_Suniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMLSub: random_Suniform_seq - return array of random numbers from uniform distribution

"random_get_seed" 1

AMLSub: random_Srandom_get_seed - return current seed for random numbers

"random_set_seed" 1

AMLSub: random_Srandom_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: rdy (rdy) - print listener ready message:

"read" 9
  remark: a user's access mode to a segment may be any combination of "reva" (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_$resetread - delete unuseful read-ahead collected for an ioname
  SSSsub: ios_$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: rdy (rdy) - print listener ready message:
  SSScmd: rdy_off (rdf) - turn off listener ready messages
  SSScmd: rdy_on (rin) - turn on listener ready messages

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

"ready_on" 1
  SSScmd: rdy_on (rin) - 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 (qq) - 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

```

SSSub: area_redef - change the length of an area:

"reduce" 1

SSSub: 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
SSCmd: get_pathname (gpn) - return absolute path name for given reference name
SSCmd: initiate (in) - make a file known by a given reference name
SSCmd: terminate (tm) - terminate a reference name
SSCmd: terminate_refname (tmr) - terminate a segment given by reference name
SSSub: hcs_sfs_get_fat_name - return reference names for a segment from KST
SSSub: hcs_sfs_get_ptr - return pointer to segment given reference name
SSSub: hcs_sfs_initiate - make a file known by reference name in KST, return pointer
SSSub: hcs_sfs_initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSub: hcs_sfs_make_seg - create file branch given absolute path name and make known by given reference name in KST
SSSub: hcs_sfs_terminate_file - terminate all reference names for branch from KST, make segment unknown
SSSub: hcs_sfs_terminate_name - terminate one reference name for segment from KST
SSSub: hcs_sfs_terminate_name - terminate a null reference name for a segment from KST
SSSub: hcs_sfs_terminate_seg - terminate all reference names for a segment from KST, make segment unknown
SSSub: ios_sseek - set reference pointers associated with an iorname
SSSub: ios_stell - return value of reference pointer associated with an iorname
SSSub: term_srefname - same as term_ for segment given by reference name

"refname" 1

SSSub: term_srefname - same as term_ for segment given by reference name

"relate" 1

SSSub: hcs_sset_relatav - 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
SSSub: expand_path_ - expand relative path name into absolute path name

"release" 1

SSCmd: release - release machine conditions saved by "hold"

"remark" 65

remark: Multics (multiplexed information and computing service) is a time-sharing system on the OS645
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, whose 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 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 "read", "write", "append", or "null"
remark: a common access control list (CACLI) 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_init>projectname"
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" (ATM 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 output 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 "SSSaf" 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 (fn) " 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_toRepeat_call - return to given stack frame and repeat call
"repeat-call" 1
AMLCmd: return_toRepeat_call - return to given stack frame and repeat call
"replace" 1
SSSub: hcs-$acl_replace - replace the entire ACL or CACL of a branch
"request" 22
Remark: a DCW (data control word) requests the GPC to perform some action on a device channel
SSSCmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSSCmd: dpunch (dph) - request delayed punching of a file on high-speed punch
AMLCmd: eplbsa - request delayed assembly of programs in the eplbsa language
SSSCmd: logout - request process destruction and session termination
SSSCmd: new_proc - request user control to create a new user process
SSSCmd: plld: request delayed compile of programs in the PL1 language
SSSub: acm-$wakeUp_int_secs - request wakeUp over event channel after interval in seconds
SSSub: acm-$wakeUp_interval - request wakeUp over event channel after interval in microseconds
SSSub: acm-$wakeUp_time - request wakeUp over event channel at time given in microseconds
SSSub: hcs-$assign - request GIM to assign and initialize device channel
SSSub: hcs-$list_change - request GIM to change DCW list
SSSub: hcs-$list_connect - request GIM to start device channel on DCW list
SSSub: hcs-$safety - request GIM to stop device channel and safety DCW list
SSSub: hcs-$set_alarm - request wakeUp at given time in microseconds
SSSub: hcs-$set_timer - request wakeUp or fault after given CPU time interval
SSSub: hcs-$unassign - request GIM to un-assign device channel
SSSub: ios-$order - issue request to outer module associated with an ioname
AMLCmd: ipc-$mask_ev_calls - request ipc_$block not to interrogate event-call channels
AMLCmd: ipc-$set_call_prio - request ipc_$block to interrogate event-call channels before event-wait channels
AMLCmd: ipc-$set_wait_prio - request ipc_$block to interrogate event-wait channels before event-call channels
AMLCmd: ipc-$unmask_ev_calls - request ipc_$block to interrogate event-call channels
"resetread" 1
SSSub: ios-$resetread - delete unused read-ahead collected for an ioname
"resetwrite" 1
SSSub: ios-$resetwrite - delete unused write-behind collected for an ioname
"respect" 1
SSSub: hcs-$fs-set_mode - return access of user with respect to segment
"restart" 1
SSSCmd: start - restart execution at saved machine conditions
"return" 75
Remark: each process has a call-save-return stack segment for each ring (see MPX RefData 2.3)
Remark: MPX RefData 2.3 gives the formats of the stack and argument lists and the call-save-return sequences
SSSub: af: get_pathname (gpn) - return absolute path name for given reference name
SSSub: af: pd - return absolute path name of process directory
AMLCmd: return_to - return to a given stack frame
AMLCmd: return_toRepeat_call - return to given stack frame and repeat call
SSSub: alloc - allocate storage in given area, return pointer
SSSub: clock - return calendar clock time in microseconds
SSSub: com_err-$check_fs_errcode - return character string for system error code
SSSub: cu_$arg_count - return number of arguments of calling procedure
SSSub: cu_$arg_list_ptr - return pointer to calling procedure's argument list
SSSub: cu_$level_ptr - return pointer to character string argument
SSSub: cu_$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_ll \$tag_star - return user's access control name
SSSsub: get_dir - return absolute path name of process directory
SSSsub: get_process_ll - return user process id
SSSsub: get_ring - return the current ring number:
SSSsub: hcs_acl_list - return contents of the ACL or CACL or CACL of a branch
SSSsub: hcs_scravi_out_data - return machine conditions at last fault in ring 0
SSSsub: hcs_sfs_get_brackets - return ring brackets and access for a segment from KST
SSSsub: hcs_sfs_get_dir_name - return absolute path name of parent directory for a segment
SSSsub: hcs_sfs_get_mode - return access of user with respect to segment
SSSsub: hcs_sfs_get_path_name - return absolute path name of parent directory and entry name of a segment
SSSsub: hcs_sfs_get_ref_name - return reference names for a segment from KST
SSSsub: hcs_sfs_get_seg_ptr - return pointer to segment given reference name
SSSsub: hcs_sfs_search_get_wdir - return absolute path name of current working directory
SSSsub: hcs_sget_cur_status - return SIM position in DCW list
SSSsub: hcs_sget_entry_name - return entrypoint name given offset in gate segment
SSSsub: hcs_sget_page_frame - return information about last 200 page faults
SSSsub: hcs_sinitiate - make a file known by reference name in KST, return pointer
SSSsub: hcs_sinitiate_count - make a file known by reference name in KST, return pointer
SSSsub: hcs_smake_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_sproc_info - return process id, process group id, process directory name, ring number
SSSsub: hcs_squota_get - return record quota, current use, and cumulative use for directory branch
SSSsub: hcs_sstar_list - search a directory for entry names matching template, return long information
SSSsub: hcs_sstatus_long - return information from directory entry about a branch or link
SSSsub: hcs_sstatus_minf - return bit count and type for a branch or link
SSSsub: hcs_sstatus_minv - return bit count and type for a segment
SSSsub: hcs_susage_values - return cumulative page waits and CPU time used
SSSsub: loa_sfs - format a message and return character string
SSSsub: loa_srsnrl - format a message and return character string, no newline
SSSsub: los_sgetdelim - return current read delimiters for ioname
SSSsub: los_sgetmode - return binary encoding of mode of attachment for ioname
SSSsub: los_sgetsize - return current element size for an ioname
SSSsub: los_stell - return value of reference pointer associated with an ioname
SSSsub: los_suptata - return status for asynchronous transaction on an ioname
AMSsub: ipc_schn_1 - return name of this validation level's event channel 1
AMSsub: ipc_schn_2 - return name of this validation level's event channel 2
SSSsub: parse_file - scan character file and return indices of next break or symbol
SSSsub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: parse_file_exponential - return random number from exponential distribution with mean 1
AMSsub: random_exponential_seq - return array of random numbers from exponential distribution
AMSsub: random_snormal - return random number from normal distribution
AMSsub: random_snormal_ant - return random number from normal distribution
AMSsub: random_snormal_seq - return array of random numbers from normal distribution
AMSsub: random_snormal_get_seed - return current seed for random numbers
AMSsub: random_suniform - return random number from uniform distribution
AMSsub: random_suniform_ant - return random number from antithetic uniform distribution
AMSsub: random_suniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMSsub: random_suniform_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_torepeat_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

 "reva" 1
 remark: a user's access mode to a segment may be any combination of "reva" (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 (hardware) supervisor
 remark: each process has a call-back-return stack segment for each ring (see WP4 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_scravl_out_data - return machine conditions at last fault in ring 0
 SSSsub: hcs_sfs_get_brackets - return ring brackets and access for a segment from KST
 SSSsub: hcs_sproc_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\$roundshftb - round a number to multiple of 2**n, divide by 2**n

 "rndshftb36" 1
 AMLsub: roundb\$roundshftb36 - 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\$roundshftb - round a number to multiple of 2**n, divide by 2**n
 AMLsub: roundb\$roundshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
 AMLsub: roundb\$roundshftb36 - 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\$roundshftb - round a number to multiple of 2**n, divide by 2**n
 AMLsub: roundb\$roundshftb36 - round a number to multiple of 36, round to multiple of 2**n, divide by 2**n
 AMLsub: roundb\$roundshftb36 - round a number to multiple of 36, round to multiple of 2**n

"roundb36" 1
 ANIsb: roundb\$roundb36 - round a number to multiple of 36, round to multiple of 2**n
"rs" 1
 SSSsub: i0a_srs - format a message and return character string
"rsnl" 1
 SSSsub: i0a_srsnl - 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
 ANIcmd: 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: binarchive (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: changevdir (cvt) - change current working directory
SSSCmd: console_output (cb) - attach "user_output" to the console ioname "user_1/5"
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 (fa) - delete item from an ACL
SSSCmd: deletetac1 (dc) - delete item from a CACL
SSSCmd: deletedir (dd) - delete a directory branch and all files contained in it
SSSCmd: deleteforce (lf) - delete a file branch; set write access if necessary
SSSCmd: deletename (fn) - delete entry name from a file branch
SSSCmd: dprint (dp) - request delayed printing of a file on high-speed printer
SSSCmd: dpunch (dpr) - 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 (hnu) - print out current number of users and maximum number
SSSCmd: initiate (in) - make a file known by a given reference name
SSSCmd: locall - call "ios_" with given arguments
SSSCmd: lmode - 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: listnames (ln) - print out entry names for files in a directory


```

SSSCmd: listtotal (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: move (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 (ptt) - 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 iconame attachments for process
SSSCmd: print_link_info (pli) - print linkage and endpoint information for an object segment
SSSCmd: print_object_map (pom) - print object map for an object segment
SSSCmd: print_homedir (phd) - print absolute path name of home directory
SSSCmd: printvdir (pvi) - 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 (rof) - turn off listener ready messages
SSSCmd: ready_on (ron) - 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: setacl (sc) - add item to CACL for a directory
SSSCmd: set_homedir (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

```

"SSSub" 160

```

remark: SSS subroutines are marked "SSSub" in this index; consult the SSS section of the MPM
SSSub: acm$$sleep_int_secs - block process for time interval in seconds
SSSub: acm$$sleep_interval - block process for time interval in microseconds
SSSub: acm$$sleep_time - block process until time given in microseconds
SSSub: acm$$wakeup_int_secs - request wakeup over event channel after interval in seconds
SSSub: acm$$wakeup_interval - request wakeup over event channel after interval in microseconds
SSSub: acm$$wakeup_time - request wakeup over event channel at time given in microseconds
SSSub: active_fnc_arr - write error message on "user_output" and signal condition "active_function_error"
SSSub: alloc - allocate storage in given area, return pointer
SSSub: area - initialize an area
SSSub: area$redef - change the length of an area
SSSub: check_star - check whether path name contains a star
SSSub: clock - return calendar clock time in microseconds
SSSub: com_err - format system error message and write on "user_output"
SSSub: com_err$check_fs_errcode - return character string for system error code
SSSub: com_err$notify - format non-system error message and write on "user_output"
SSSub: condition - establish handler for a condition
SSSub: cu_sarg_count - return number of arguments of calling procedure
SSSub: cu_sarg_list_ptr - return pointer to calling procedure's argument list
SSSub: cu_sarg_ptr - return pointer to character string argument

```

SSSSub: cu_scl - call current listener to re-enter; command level
SSSSub: cu_scp - call current command processor to execute command line
SSSSub: cu_sgen_call - call procedure given by pointer with given argument list
SSSSub: cu_sgrov_stack_frame - increase size of calling procedure's stack frame
SSSSub: cu_slevel_get - return current validation level
SSSSub: cu_slevel_set - set validation level
SSSSub: cu_sptr_call - call procedure given by pointer with given arguments
SSSSub: cu_sset_cl - refine current listener
SSSSub: cu_sset_cop - refine current command processor
SSSSub: cu_sshrink_stack_frame - reduce size of calling procedure's stack frame
SSSSub: cu_sstack_frame_ptr - return pointer to stack frame of calling procedure
SSSSub: cu_sstack_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
SSSSub: date_time - convert calendar clock time to character string date and time
SSSSub: date_time_sfstime - 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: free - 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_star - return user's access control name with instance tag a star
SSSSub: get_ptr - 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_sacl_add - add or change items on the ACL or CACL of a branch
SSSSub: hcs_sacl_add1 - add or change single ACL or CACL item for a branch
SSSSub: hcs_sacl_delete - delete all or part of the ACL or CACL of a branch
SSSSub: hcs_sacl_list - return contents of the ACL or CACL of a branch
SSSSub: hcs_sacl_replace - replace the entire ACL or CACL of a branch
SSSSub: hcs_sappend_branch - create a file branch entry in parent directory
SSSSub: hcs_sappend_branchx - create directory or file branch entry in parent directory and add user to ACL
SSSSub: hcs_sassign_link - create a link entry in parent directory
SSSSub: hcs_sassign - request GIM to assign and initialize device channel
SSSSub: hcs_sassign_linkage - allocate storage in linkage section
SSSSub: hcs_schname_file - add and delete entry names on branch or link entry
SSSSub: hcs_schname_seg - add and delete entry names on branch or link entry
SSSSub: hcs_scrawl_out_data - return machine conditions at last fault in ring 0
SSSSub: hcs_sdel_dir_tree - delete all branches inferior to a directory
SSSSub: hcs_sdelentry_file - delete branch or link entry from parent directory
SSSSub: hcs_sdelentry_seg - delete entry for a segment from parent directory
SSSSub: hcs_sfs_get_brackets - return ring brackets and access for a segment from KST
SSSSub: hcs_sfs_get_dir_name - return absolute path name of parent directory for a segment
SSSSub: hcs_sfs_get_mode - return access of user with respect to segment
SSSSub: hcs_sfs_get_path_name - return absolute path name of parent directory
SSSSub: hcs_sfs_get_ref_name - return reference name for a segment from KST
SSSSub: hcs_sfs_get_seg_ptr - return pointer to segment given reference name
SSSSub: hcs_sfs_move_file - move contents from one file branch to another
SSSSub: hcs_sfs_move_seg - move contents from one segment to another
SSSSub: hcs_sfs_search_get_vdir - return absolute path name of current working directory
SSSSub: hcs_sfs_search_set_vdir - change current working directory
SSSSub: hcs_sget_cur_status - return GIM position in DCW list
SSSSub: hcs_sget_entry_name - return entrypoint name given offset in gate segment


```

SSSsub: ip$svwrite_ptr - write on ioname "user_output"
SSSsub: ios$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_sparse_file_cur_line - return indices of current line of character file
SSSsub: parse_file_sparse_file_init - initialize parse_file with given segment name
SSSsub: parse_file_sparse_file_ptr - initialize parse_file with pointer to segment
SSSsub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol
SSSsub: parse_file_sparse_file_set_break - declare break characters to parse_file
SSSsub: parse_file_sparse_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_snomakeunknown - same as term for segment given by pointer, but do not terminate
SSSsub: term_srefname - same as term for segment given by reference name
SSSsub: term_sseg_ptr - same as term for segment given by pointer
SSSsub: ti_sfindata - turn scratch segment into data segment
SSSsub: ti_sfinobj - turn scratch segment into object segment
SSSsub: ti_sgetseg - 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 GEM to stop device channel and safety DCW list

"same" 3
SSSsub: term_snomakeunknown - same as term for segment given by pointer, but do not terminate
SSSsub: term_srefname - same as term for segment given by reference name
SSSsub: term_sseg_ptr - same as term for segment given by pointer

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

"saver" 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

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

"sc" 1
SSScmd: setcaci (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_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

"sc1" 1
SSScmd: set_com_line (sc1) - set maximum expanded command line length

"scratch" 3
SSSsub: ti_$indata - turn scratch segment into data segment
SSSsub: ti_$inobj - turn scratch segment into object segment
SSSsub: ti_$setseg - 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_Smake_ptr - return pointer to entrypoint given segment name and entrypoint name (searches and makes known)
SSSsub: hcs_Sstar_ - search a directory for entry names matching template
SSSsub: hcs_Sstar_list_ - search a directory for entry names matching template, return long information

"second" 2
SSSsub: acm-$sleep_int_secs - block process for time interval in seconds
SSSsub: acm-$wake_up_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_Sassign_linkate - 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_$random_set_seed - return current seed for random numbers
AMLsub: random_$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 (hardware) 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: 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_archive - 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 (pdm) - 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_\$delete_entry_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_bits - 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_sparse_file_init - initialize parse_file with given segment name
 SSSsub: parse_file_sparse_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_\$vakeup - 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 ReData 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$block - set bit count on file branch, start from current length
SSScmd: deleteforce (lf) - delete a file branch; set write access if necessary
AMlcmd: entry_usage$clear - set endpoint usage counts to 0
SSScmd: lmode - 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 idirectory
SSSub: cu_$level-set - set validation level
SSSub: hcs_$list_size - set size of SIM PCW list for device channel
SSSub: hcs_$set_bc - set bit count in file branch entry
SSSub: hcs_$set_consistsw - set consistency switch in branch entry
SSSub: hcs_$set_copysw - set copy switch in branch entry
SSSub: hcs_$set_relatasw - set relate switch in branch entry
SSSub: ios_$readsync - set read synchronization mode for an ioname
SSSub: ios_$seek - set reference pointers associated with an ioname
SSSub: ios_$setdelim - set current read delimiters for an ioname
SSSub: ios_$setsize - set current element size for an ioname
SSSub: ios_$worksync - set workspace synchronization mode for an ioname
SSSub: ios_$writesync - set write synchronization mode for an ioname
AMlsub: random_$random_set_seed - set seed for random numbers
SSSub: ti_$setseg - set up a scratch segment

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

"set_bc" 1
SSSub: 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
SSSub: cu_$set_cl - define current listener

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

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

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

```

"set_cp" 1
SSSub: cu_sset_cp = define current command processor

"set_relatelv" 1
SSSub: hcs_sset_relatelv = set relate switch in branch entry

"set_timer" 1
SSSub: hcs_sset_timer = request wakeup or fault after given CPU time interval

"set_vait_prior" 1
AHSsub: ipc_sset_vait_prior = request ipc_block to interrogate event_vait channels before event-call channels

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

"setcaci" 1
SSScmd: setcaci (sc) = add item to CACL for a directory

"setdelim" 1
SSSub: ios_ssetdelim = set current read delimiters for an ioname

"sethomedir" 1
SSScmd: sethomedir (shd) = set path name of home idrectory

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

"setsize" 1
SSSub: ios_ssetsize = 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 GZ645

"shd" 1
SSScmd: sethomedir (shd) = set path name of home idrectory

"shrink_stack_frame" 1
SSSub: cu_shrink_stack_frame = reduce size of calling procedure's stack frame

"sign" 1
SSSub: 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: MPH RefData 2.2.1 lists all condition names which the system may signal
SSScmd: program_interrupt (pi) = signal the condition "program_interrupt"
SSSub: active_fnc_err_ = write error message on "user_output" and signal condition "active_function_error"
SSSub: 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
SSSub: signal_ = signal the occurrence of a condition; call its handler

```


"single" 1
SSSub: hcs-\$acl_add1 - add or change single ACL or CACL item for a branch

"size" 6
SSSub: cu-\$grov_stack_frame - increase size of calling procedure's stack frame
SSSub: cu-\$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSub: cu-\$stack_frame_size - return size of calling procedure's stack frame
SSSub: hcs-\$list_size - set size of SIM DCW list for device channel
SSSub: ios-\$getsize - return current element size for an ioname
SSSub: ios-\$setsize - set current element size for an ioname

"sleep_int_secs" 1
SSSub: acm-\$sleep_int_secs - block process for time interval in seconds

"sleep_interval" 1
SSSub: acm-\$sleep_interval - block process for time interval in microseconds

"sleep_time" 1
SSSub: 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 GJC to perform some action on a device channel

"sort" 1
SSCmd: 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
SSCmd: dprint (dp) - request delayed printing of a file on high-speed printer.
SSCmd: dpunch (dpn) - request delayed punching of a file on high-speed punch

"st" 1
SSCmd: 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
AMCmd: return_to - return to a given stack frame
AMCmd: return_to-repeat_call - return to given stack frame and repeat call
SSSub: cu-\$grov_stack_frame - increase size of calling procedure's stack frame
SSSub: cu-\$shrink_stack_frame - reduce size of calling procedure's stack frame
SSSub: cu-\$stack_frame_ptr - return pointer to stack frame of calling procedure
SSSub: cu-\$stack_frame_size - return size of calling procedure's stack frame

"stack_frame_ptr" 1
SSSub: cu-\$stack_frame_ptr - return pointer to stack frame of calling procedure

"stack_frame_size" 1
SSSub: cu-\$stack_frame_size - return size of calling procedure's stack frame

"standard" 1

```

Remark: MPH RefData 2.7 describes the system standard search rules used by the linker
"star" 2
SSSub: check_star_ " check whether path name contains a star
SSSub: get_group_id_star " return user's access control name with instance tag a star
"star" 1
SSSub: hcs_star_ " search a directory for entry names matching template
"star_list_" 1
SSSub: hcs_star_list_ " search a directory for entry names matching template
"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_block " check bit count on file branch, start from current length
SSCmd: start " restart execution at saved machine conditions
SSSub: hcs_list_connect " request GIM to start device channel on DCW list
"status" 4
SSCmd: status (st) " print out status information from directory entry for a branch
SSSub: hcs_get_status " return GIM status for device channel
SSSub: hcs_status " return information from directory entry about a branch or link
SSSub: ios_supstate " return status for asynchronous transaction on an ioname
"status_long" 1
SSSub: hcs_status_long " return all information from directory entry about a branch or link
"status_minf" 1
SSSub: hcs_status_minf " return bit count and type for a branch or link
"status_mins" 1
SSSub: hcs_status_mins " return bit count and type for a segment
"stop" 1
SSSub: hcs_safety " request GIM to stop device channel and safety DCW list
"storage" 3
SSSub: alloc_ " allocate storage in given area, return pointer
SSSub: free_ " free allocated storage in an area
SSSub: hcs_sasign_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
SSSub: com_err_check_errcode_ " return character string for system error code
SSSub: cu_sarg_ptr " return pointer to character-string argument
SSSub: cv_bin_ " convert binary to character string, any base
SSSub: cv_bin_dec " convert binary to character string, base 10
SSSub: cv_bin_oct " convert binary to character string, base 8
SSSub: cv_dec_ " convert character string to binary, base 10
SSSub: cv_dec_check_ " convert character string to binary, base 10; return error code
SSSub: cv_oct_ " convert character string to binary, base 8
SSSub: cv_oct_check_ " convert character string to binary, base 8; return error code
SSSub: date_time_ " convert calendar clock time to character string date and time
SSSub: date_time_sfstime " convert file system time to character string date and time

```

SSSSub: loa_sfs - format a message and return character string
SSSSub: loa_srsnal - 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 MPH
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 MPH

"supervisor" 1

Remark: the gate segment "hcs_" allows the user to call into the ring 0 (hardware) supervisor

"switch" 4

Remark: users make calls to the input-output switch "ios_" to read and write on iónames
SSSSub: hcs_\$set_consistsw - set consistency switch in branch entry
SSSSub: hcs_\$set_copysw - set copy switch in branch entry
SSSSub: hcs_\$set_relatasw - set relate switch in branch entry

"symbol" 6

AMLcmd: createsyntab (cst) - create symbol table files for pl1-language programs (used with "debug")
AMLcmd: debug (db) - symbolic interactive debugging aid
AMLcmd: printsyntab (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_sparse_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_\$writsync - 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 output 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

Index for Multics - 36 words beginning with t.

"Teletype" 1
 remark: "quit" is signalled after a "quit signal" (ATTN 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
 SSSub: get_group_ii_\$tag_star - return user's access control name with instance, tag a star

"tag_star" 1
 SSSub: get_group_ii_\$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
 SSSub: ios_\$tell - return value of reference pointer associated with an iofname

"template" 2
 SSSub: hcs_\$star - search a directory for entry names matching template
 SSSub: hcs_\$star_list - search a directory for entry names matching template, return long information

"term" 4
 SSSub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
 SSSub: term_\$nomakeunknown - same as term_ for segment given by pointer, but do not terminate
 SSSub: term_\$refname - same as term_ for segment given by reference name
 SSSub: 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.
 SSCcmd: bsys - interactive editor and terminal interface for programming in the basic language

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

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

"terminate_name" 1
 SSSub: 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 KSI

"terminate_refname" 1
 SSSCmd: terminate_refname (tmt) - terminate a segment given by reference name

"terminate_seg" 1
 SSSSub: hcs_\$terminate_seg - terminate all reference names for a segment from KSI. make segment unknown

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

"termination" 1
 SSSCmd: logout - request process instruction and session termination

"test" 1
 AMICmd: adjuststest - check bit count on file branch

"test_block" 1
 AMICmd: adjuststest_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 NPM 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_\$findbj - 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_\$ftime - 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

"tmt" 1
 SSSCmd: terminate_refname (tmt) - 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_\$slowalt - wait for transactions on workspace asynchronous ioname
 SSSSub: ios_\$supstate - 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 (rdf) - turn off listener ready messages
 SSScmd: ready_on (rin) - turn on listener ready messages
 SSSsub: ti_\$indata - turn scratch segment into data segment
 SSSsub: ti_\$inobj - 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: MPH RefData 1.4 describes the input-output system and device and output module types
 SSSsub: hcs-\$status_ninf - return bit count and type for a branch or link
 SSSsub: hcs-\$status_nins - return bit count and type for a segment
 AMLsub: ipc-\$decl_ev_call_chh - make event channel into event-call type
 AMLsub: ipc-\$decl_ev_wait_chh - 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: MPH RefData 2.5 describes canonical form, typing conventions, and erase, kill, and escape characters

Index for Multics. - 27 words beginning with u.

"ul" 1
SSCmd: unlink (ul) - delete a link entry from a directory
"unassign" 1
SSSub: hcs_sunassign - request GIM to unassign a device channel
"unclaimed_signal" 1
remark: "signal_" signals the "unclaimed_signal" condition if it cannot find a handler for a condition
"undeclare" 1
SSSub: parse_file_sparse_file_unset_break - underscore break characters for parse_file_
"uniform" 4
AMSub: random_uniform - return random number from uniform distribution
AMSub: random_uniform_ant - return random number: from antithetic uniform distribution
AMSub: random_uniform_ant_seq - return array of random numbers from antithetic uniform distribution
AMSub: random_uniform_seq - return array of random numbers from uniform distribution
"uniform_ant" 1
AMSub: random_uniform_ant - return random number: from antithetic uniform distribution
"uniform_ant_seq" 1
AMSub: random_uniform_ant_seq - return array of random numbers from antithetic uniform distribution
"uniform_seq" 1
AMSub: random_uniform_seq - return array of random numbers from uniform distribution
"unique" 2
SSSub: unique_bits - return unique bit string
SSSub: unique_chars - convert bit string into unique character string
"unique_bits" 1
SSSub: unique_bits - return unique bit string
"unique_chars" 1
SSSub: unique_chars - convert bit string into unique character string
"unlink" 1
SSCmd: unlink (ul) - delete a link entry from a directory
"unlock" 1
AMCmd: mailunlock - unlock a locked mailbox
"unmask_ev_calls" 1
AMSub: ipc_sunmask_ev_calls - request ipc_block to interrogate event=call channels
"unsnap" 1
SSSub: term_ - remove segment's linkage section, unsnap linkage pointers to segment, and terminate segment
"unti
1" 3
SSSub: acm_sleep_time - block process until time given in microseconds
AMSub: hcs_block - block process until next wakeup
AMSub: ipc_block - block process until given event
"unused" 2

```

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_usageclear - set endpoint 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 (q) - 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: createsymbtab (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 (hardware) 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 "reva" (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 (hnu) - 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_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 is 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_id" 1
SSSCmd: console-output (co) - attach "user_output" to the console ioname "user_1/5"

"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
SSSub: active_inc_err_ - write error message on "user_output" and signal condition "active_function_error"
SSSub: com_err_ - format system error message and write on "user_output"
SSSub: com_err_notify_ - format non-system error message and write on "user_output"
SSSub: ioa_ - format a message and write it on "user_output"
SSSub: ioa_notify_ - format a message and write it on "user_output", no newline
SSSub: ios_write_ptr_ - write on ioname "user_output"
A1Sub: write_list_ - write variables on "user_output"
```

Index for Multics - 4 words beginning with v

"valid" 2
 AMLEnd: ascii_check - Check all characters of a file for valid ASCII
 AMLEnd: 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
 AMSub: ipc-\$chn_1 - return name of this validation level's event channel 1
 AMSub: ipc-\$chn_2 - return name of this validation level's event channel 2
"value" 1
 SSSsub: ios_\$stell - return value of reference pointer associated with an ioname
"variable" 2
 AMSub: read_list_ - read in free-format variables from "user_input"
 AMSub: 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
SSSub: hcs-\$usage_values - return cumulative page-waits and CPU time used
SSSub: ios-\$iowait - wait for transactions on work-space asynchronous iohame
AMSub: ipc-\$decl_ev_wait_chn - make event channel into event-wait type (default)
AMSub: ipc-\$set_call_prior - request ipc_block to interrogate event-call channels before event-wait channels
AMSub: 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
SSSub: acm-\$wakeup_int_secs - request wakeup over event channel after interval in seconds
SSSub: acm-\$wakeup_interval - request wakeup over event channel after interval in microseconds
SSSub: acm-\$wakeup_time - request wakeup over event channel at time given in microseconds
AMSub: hcs-\$block - block process until next wakeup
SSSub: hcs-\$set_alarm - request wakeup at given time in microseconds
SSSub: hcs-\$set_timer - request wakeup or fault after given CPU time interval
AMSub: hcs-\$wakeup - send inter-process wakeup to process with given id
AMSub: ipc-\$drain_chn - drain event channel of any pending wakeup

"wakeup_int_secs" 1

SSSub: acm-\$wakeup_int_secs - request wakeup over event channel after interval in seconds

"wakeup_interval" 1

SSSub: acm-\$wakeup_interval - request wakeup over event channel after interval in microseconds

"wakeup_time" 1

SSSub: acm-\$wakeup_time - request wakeup over event channel at time given in microseconds

"wd" 1

SSSub: 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

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

"whoami" 1

AMSub: 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 GPC to perform some action on a device channel
SSSub: 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: chgenvdir (cwd) - change current working directory
SSScmd: printvdir (pwd) - print absolute path name of current working directory
SSS_at: wd - return absolute path name of current working directory
SSSub: hcs_sfs_search_get_wdir - return absolute path name of current working directory
SSSub: hcs_sfs_search_set_wdir - change current working directory

"workspace" 2

SSSub: ios_slovalt - wait for transactions on workspace asynchronous ioname
SSSub: ios_sworksync - set workspace synchronization mode for an ioname

"worksync" 1

SSSub: ios_sworksync - set workspace synchronization mode for an ioname

"write" 16

remark: a user's access mode to a segment may be any combination of "read", "execute", "write", "append", or "null"
remark: users make calls to the input-output switch "ios_" to read and write on ionames
SSScmd: deleteforce (if) - delete a file branch; set write access if necessary
SSSub: active_func_err - write error message on "user_output" and signal condition "active_function_error"
SSSub: com_err - format system error message and write on "user_output"
SSSub: com_err_notify - format non-system error message and write on "user_output"
SSSub: ioa - format a message and write it on "user_output"
SSSub: ioa_sioa_stream - format a message and write it on given ioname
SSSub: ioa_sioa_stream_nhl - format a message and write it on given ioname, no newline
SSSub: ioa_sioa_nhl - format a message and write it on "user_output", no newline
SSSub: ios_deletewrite - delete unused write-behind collected for an ioname
SSSub: ios_write - write on an ioname
SSSub: ios_write_ptr - write on ioname "user_output"
SSSub: ios_writesync - set write synchronization mode for an ioname
SSSub: 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

SSSub: ios_swrite_ptr - write on ioname "user_output"

"writesync" 1

SSSub: ios_swritesync - 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
THV 2/70 */

dcl (path, path1) char (168) aligned.
  ap ptr,
  pp ptr,
  p ptr,
  p1 ptr,
  p2 ptr,
  bchr char (32) based (ap) unaligned,
  q ptr,
  word char (32) aligned,
  al fixed bin,
  ec fixed bin,
  bk fixed bin,
  nw fixed bin init (0),
  nx fixed bin init (0),
  noo fixed bin init (0),
  alphabet char (52) aligned int static init
  ("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"),
  letter char (1) aligned,
  NULC char (1) aligned,
  v(0: 26) fixed bin init ((27)0),
  wcount(0: 26) fixed bin init ((27)0),
  (oi, ti) fixed bin,
  ci fixed bin,
  cc fixed bin,
  lno fixed bin,
  swap fixed bin init (0),
  buffer char (300) aligned,
  1 pchrx based (pp) aligned,
  2 pchr char (262143),
  out char (32) aligned,
  outl fixed bin,
  ofile char (168) aligned,
  st bit (72) aligned,
  statb ptr,
  1 statx based (statp) aligned,
  2 fer fixed bin,
  (i, j) fixed bin,
  lth fixed bin;

dcl 1 wordseg based (p1) aligned,
  2 ws (1000),
  3 wd char (32),
  3 wx fixed bin,
  3 ct fixed bin,
  3 lp fixed bin,
  3 zp fixed bin;

dcl 1 lineseg based (p2) aligned,
  2 lst (1000),
  3 the word */
  fore ptr */
  number of times seen */
  index in lineseg for chain */
  index of last entry of chain */

/* path names */
/* ptr to arqlist */
/* ptr to input */
/* ptr to suppressfile */
/* ptr to wordseg */
/* ptr to lineseg */
/* pickup for args */
/* ptr to word */
/* temporary - word of interest */
/* lth of arg */
/* error code */
/* 1 if break char */
/* number of ents in wpriseg */
/* number of ents in lineseg */
/* number of suppressed wds */
/* alphabet */
/* letter which begins wpr1 */
/* pointers to word lists */
/* count of real words */
/* temps */
/* index of start of wpr1 */
/* length of word */
/* line number */
/* counter in sort */
/* print line buffer */
/* access structure for file */
/* streamname */
/* filename */
/* ios status */
/* ptr to st */
/* status breakdown */
/* temps */
/* bit length of input */
/* scratch seg for words */
/* the word */
/* fore ptr */
/* number of times seen */
/* index in lineseg for chain */
/* index of last entry of chain */
/* scratch seg for crossreference */

```



```

end;
nw = nv + 1;
vd(nv) = word;
ct(nv) = 1;
wnx(nv) = wnx(v(i));
wnx(v(i)) = nv;
go to lp1;

endsupr; call hcs_terminate_seq (p, o, ec);
noo = nv;

/* now set up for reading file */
nosupr; call parse_file_sparse_file_init (path1, op, ec); /* Set up to read input. */
if ec = 0 then go to ar;
call parse_file_sparse_file_unset_break (">");

loop: call parse_file_sparse_file_ptr (a, cc, bk, sc); /* 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 (a -> bchr, 1, cc); /* get one word of input */
letter = substr (word, 1, 1); /* get first letter */
i = index (alphabet, letter); /* get index in "v" */
if i > 26 then j = 1 - 26; /* map caps to smalls */

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

i = v(j); /* ptr to first vd 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? */
call parse_file_sparse_file_cur_line (cl, cc); /* find line */
if ct(i) > 0 then do; /* check for dup */
if cl = 1s(zp(i))+1 then if cc = 1l(zp(i)) then go to loop;
end;
call parse_file_sparse_file_line_no (ln); /* get line number */
ct(i) = ct(i) + 1; /* Count hits on oword. */
nx = nx + 1; /* alloc crossreference */
1s(nx) = cl-1; /* Fill in ptrs to line-start & line length */
1o(nx) = cc; /* use cr on line */
1o(nx) = ln; /* 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 vd ptr to last elem */
go to loop;

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

```

```

ends;
i = vnx(i);
go to sh;

/* now sort wordseq */
endin: call loa_ ("d references to d words in ""i"", nx, nvnnoo, path1);
sort: do j = 0 to 26;
pass: i = v(j);
if i = 0 then go to nosort;
i = vnx(i);
swap = 0;
if vnx(i) = 0 then if swap = 0 then go to nosort;
else go to pass;
if wd(i) > wd(vnx(i)) then do;
word = wd(i);
wd(i) = wd(vnx(i));
wd(vnx(i)) = word;
ti = lp(i);
lp(i) = lp(vnx(i));
lp(vnx(i)) = ti;
ti = ct(i);
ct(i) = ct(vnx(i));
ct(vnx(i)) = ti;
swap = swap + 1;
end;
/* remember swap */
/* advance i */
i = vnx(i);
go to cmp;
nosort: end sort;
call loa_ ("lists sorted, begin output,");

/* now loop thru file + print index */
call expand_path_(addr, out), out1, addr (ofile), null, ec);
call ios_sattach_(out, "file_", ofile, "", st);
statp = addr (st);
ec = fer;
if ec = 0 then go to er;

do j = 0 to 26;
i = v(j);
if i = 0 then go to end2;
if vcount(j) = 0 then go to end2;
if j > 0 then letter = substr (alphabet, j, 1);
else letter = " ";
call loa_siba_stream (out, "|index for "a = "d words beginning with "a"/"/",
path1, vcount(j), letter);
if ct(i) = -1 then do;
go to and1;
end;
else call loa_siba_stream (out, ""a"" "d", wd(i), ct(i));
/* Get start of crossref chain. */
ti = lp(i);
buffer = substr (pchr, ls(ti)+1, ll(ti)-1); /* get line */
do oi = 1 to ll(ti)-1;
if substr (buffer, oi, 1) = "x" then /* change percent to o177 */
substr (buffer, oi, 1) = NUHC;
end;

```



```

call ioa_$ia_stream (out, "5x^a", buffac); /* write line out */
ti = in(ti); /* follow chain */
if ti ^= 0 then go to lp2; /* zero next ptr flag end. */
if vnx(i) ^= 0 then do;
  i = vnx(i);
  go to lp3;
end;
end1: end;
end2: end;

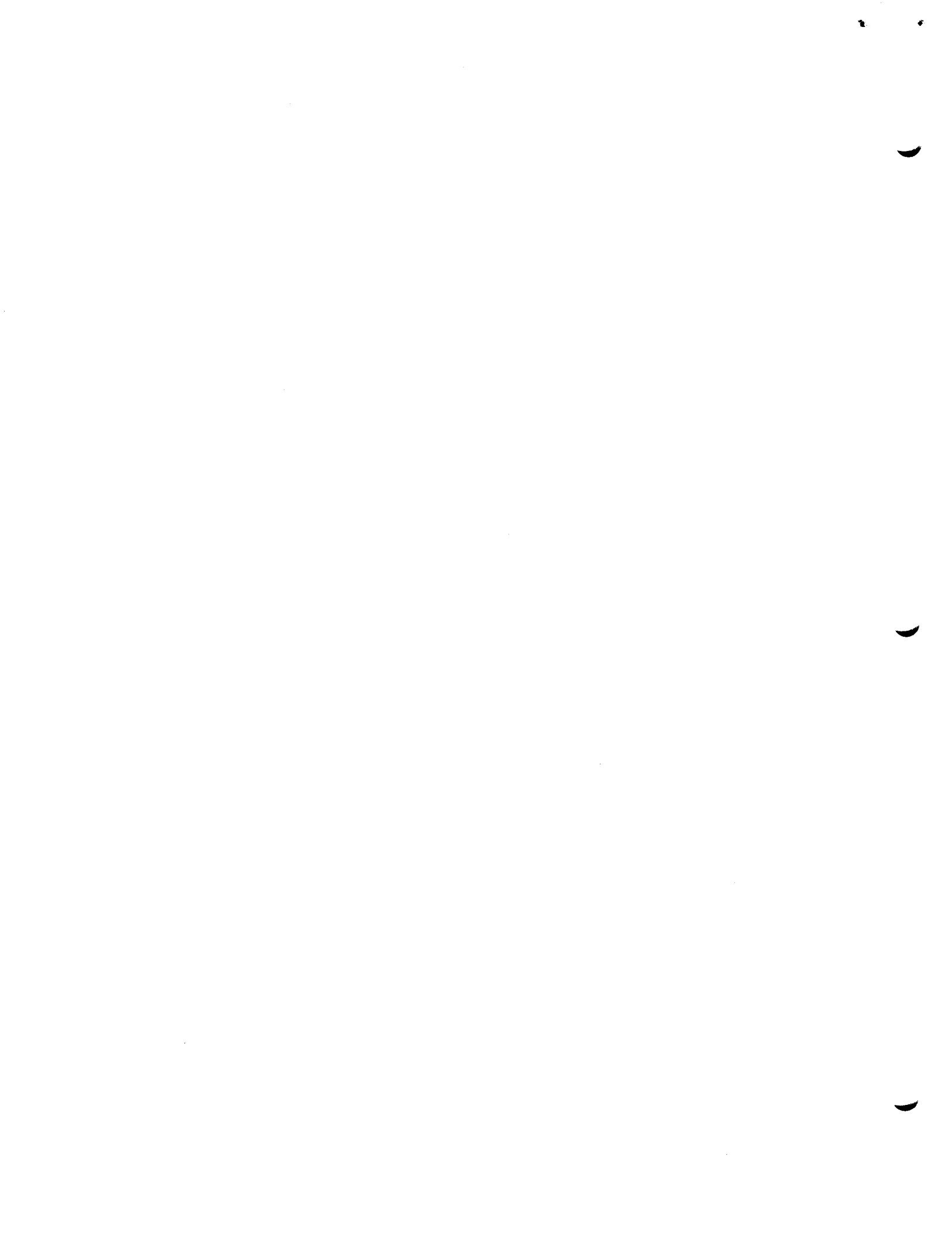
call ios_$detach (but, "", "", st);

call ioa_ ("");
call hcs_$terminate_ssg (pp, 0, ac);
call hcs_$delentry_ssg (p1, ac);
call hcs_$delentry_ssg (p2, ac);

/* Whew. */
/* Terminate input. */
/* Delete scratch. */
/* .. */

end kvoc;

```



remark: Multiplex (multiplexed information and computing service) is a time-sharing system on the G3645

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

remark: a user normally controls his multiplex process from a typewriter terminal (console)

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

remark: a "segment" is a two-dimensional address space consisting of segments

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 name 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 KSR (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: 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 "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 ">username_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 linkage stack segment for each ring (see MPM Refdata 2.3)

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: "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 274), INTERRUPT on modal 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: 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 "SSSdef" in this index; consult the SSS section of the MPM

remark: SSS subroutines are marked "SSSub" 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

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

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
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 segments
SSSCmd: bsys - interactive editor and terminal interface for programXming in the basic language
SSSCmd: calc - desk calculator with memory for functions and expressions
SSSCmd: changevdir (cvt) - change current working directory
AMLCmd: compare_ascii (cpa) - compare two character files
SSSCmd: console_output (co) - attach "user_output" to the console ioname "user_i/o"
AMLCmd: convert_360_fortran (-360f) - convert programXs in the fortran language from 360 format
SSSCmd: copy (cp) - copy a file
SSSCmd: create (cr) - create a file branch
AMLCmd: createdir (cd) - create a directory branch
AMLCmd: createsymbtab (cst) - create symbol table fileXs for pl1-language programs (used with "debug")
AMLCmd: ctss_aarchv - extract all component fileXs 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
SSSCmd: decan - desk calculator with memory
SSSCmd: delete (dl) - delete a file
SSSCmd: deleteacl (da) - delete item from an ACL
SSSCmd: deletetac1 (dc) - delete item from a CACL
SSSCmd: deletedir (dd) - delete a directory branch and all fileXs 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 (dph) - request delayed punching of a file on high-speed punch
SSSCmd: edm - context editor for character fileXs
AMLCmd: endfile - close fortran fileXs
AMLCmd: entry_usage - print out usage counts for entrypoints
AMLCmd: entry_usage\$clear - set entrypoint usage counts to 0
AMLCmd: ep1bsa - request delayed assembly of programs in the ep1bsa language
SSSCmd: exec_com (ec) - execute a series of command lineXs contained in a file
AMLCmd: extract_archive - extract all component fileXs from a CTSS archive file
SSSCmd: file_output (fo) - attach "user_output" to a given file instead of the console
SSSCmd: fortran - compile programs in the fortran language
AMLCmd: fs-chname - add, delete, or change entry nameXs on a file
SSSCmd: get_com_line (gcl) - print out current expanded command line length
SSS_af: get_pathname (gpn) - return absolute path name for given reference name
SSSCmd: getquota (qq) - 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 fileXs
SSSCmd: hold - save machine conditions after error
SSSCmd: how_many_users (hmu) - print out current number of users and maximum number
AMLCmd: ind - indent programs in the pl1 language
SSSCmd: initiate (in) - make a file known by a given reference name
SSSCmd: local - call "lps" with given arguments
SSSCmd: lmode - 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 nameXs for given fileXs in a directory
SSSCmd: listacl (la) - print out ACL for a file
SSSCmd: listacl (lc) - print out CACL for a directory
SSSCmd: listnames (ln) - print out entry nameXs for fileXs in a directory
SSSCmd: listtotals (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

AM1cmd: lower_case - convert all characters in a file to lower case
 AM1cmd: mail (ml) - send a file to another user's mailbox
 AM1cmd: mail\$unlock - unlock a locked mailbox
 SSScmd: move (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
 SSS_af: pd - return absolute path name of process directory
 AM1cmd: pht - print horizontal tab
 SSScmd: pl1 - compile programs in the pl1 language
 SSScmd: pl1d: request delayed compile of programs in the pl1 language
 AM1cmd: pl1 - print newline
 SSScmd: print (pr) - print a character file.
 SSScmd: print_attach_table (pat) - print filename attachments for process
 SSScmd: print_link_info (pli) - print linkage and endpoint information for an object segment
 SSScmd: print_object_map (pom) - print object map for an object segment
 SSScmd: printthomedir (rhi) - print absolute path name of home directory
 AM1cmd: printsymtab (pst) - print symbol table for a pl1 program
 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'
 AM1cmd: ps - print argument string
 AM1cmd: qed - programable context editor for character files
 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
 AM1cmd: return - return to a given stack frame
 AM1cmd: return_toRepeat_call - return to given stack frame and repeat call
 AM1cmd: runoff - format a character file for printing
 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: setacl (sc) - add item to CACL for a directory
 SSScmd: sethomedir (shd) - set path name of home directory
 SSScmd: start - restart execution at saveX1 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: vd - 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: 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_\$wakeupt_int_secs - request wakeup over event channel after interval in seconds
 SSSsub: acm_\$wakeupt_interval - request wakeup over event channel after interval in microseconds
 SSSsub: acm_\$wakeupt_time - request wakeup over event channel at time given in microseconds
 SSSsub: active_fnc_err - writes 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_if_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_sarg_count - return number of arguments of calling procedure
SSSSub: cu_sarg_list_ptr - return pointer to calling procedure's argument list
SSSSub: cu_sarg_ptr - return pointer to character-string argument
SSSSub: cu_scl - call current listener to re-enter command level
SSSSub: cu_scl - call current listener to re-enter command level
SSSSub: cu_sgen_call - call procedure given by pointer with given argument list
SSSSub: cu_sgrov_stack_frame - increase size of calling procedure's stack frame
SSSSub: cu_slevel_get - return current validation level
SSSSub: cu_sptr_call - call procedure given by pointer with given arguments
SSSSub: cu_sset_cl - define current listener
SSSSub: cu_sset_cp - define current command processor
SSSSub: cu_sshrink_stack_frame - reduce size of calling procedure's stack frame
SSSSub: cu_sstack_frame_ptr - return pointer to stack frame of calling procedure
SSSSub: cu_sstack_frame_size - return size of calling procedure's stack frame
SSSSub: cv_bin - convert binary to character string, any base
SSSSub: cv_bin_sdec - convert binary to character string, base 10
SSSSub: cv_bin_sdec - convert binary to character string, base 8
SSSSub: cv_dec - convert character string to binary, base 10
SSSSub: cv_dec - convert character string to binary, base 8
SSSSub: cv_oct - 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_sftime - 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: free - free allocated storage in an area
SSSSub: get_at_entry - return outer module, attached lname, and mode for given lname
SSSSub: get_group_id - return user's access control name
SSSSub: get_group_id_star - return user's access control name with instance tag a star
SSSSub: get_pair - 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_sacl_add - add or change items on the ACL or CACL of a branch
SSSSub: hcs_sacl_add1 - add or change single ACL or CACL item for a branch
SSSSub: hcs_sacl_delete - delete all or part of the ACL or CACL of a branch
SSSSub: hcs_sacl_list - return contents of the ACL or CACL of a branch
SSSSub: hcs_sacl_replace - replace the entire ACL or CACL of a branch
SSSSub: hcs_sappend_branch - create a file branch entry in parent directory and add user to ACL
SSSSub: hcs_sappend_branchx - create directory or file branch entry in parent directory, given ACL
SSSSub: hcs_sappend_link - create a link entry in parent directory
SSSSub: hcs_sassign - request SIV to assign and initialize device channel
SSSSub: hcs_sassign_linkage - allocate storage in linkage section
AMSub: hcs_sblock - block process until next wakeup
SSSSub: hcs_schname_file - add and delete entry name%$ on branch or link entry
SSSSub: hcs_schname_seg - add and delete entry name%$ on branch or link entry
SSSSub: hcs_scrawl_out_data - return machine conditions at last fault in ring 0
SSSSub: hcs_sdel_dir_tree - delete all branches inferior to a directory
SSSSub: hcs_sdelentry_file - delete branch or link entry from parent directory
SSSSub: hcs_sdelentry_seg - delete entry for a segment from parent directory
SSSSub: hcs_sfs_get_brackets - return ring brackets and access for a segment from KSR
SSSSub: hcs_sfs_get_dir_name - return absolute path name of parent directory for a segment
SSSSub: hcs_sfs_get_mode - return access of user with respect to segment
SSSSub: hcs_sfs_get_path_name - return absolute path name of parent directory and entry name of a segment

```

SSSub: hcs \$fs_get_ref_name - return reference name for a segment from KST
SSSub: hcs \$fs_get_seg_ptr - return pointer to segment, given reference name
SSSub: hcs \$fs_move_file - move contents from one file, branch to another
SSSub: hcs \$fs_move_seg - move contents from one segment to another
SSSub: hcs \$fs_search_get_wdir - return absolute path name of current working directory
SSSub: hcs \$fs_search_set_wdir - change current working directory
SSSub: hcs \$get_cur_status - return GIM position in D-W list
SSSub: hcs \$get_entry_name - return entrypoint name given offset in gate segment
SSSub: hcs \$get_page_trace - return information about last 200 page faults
SSSub: hcs \$get_status - return GIM status for device channel
SSSub: hcs \$initiate - make a file known by reference name in KST, return pointer
SSSub: hcs \$initiate_count - make a file known by reference name in KST, return pointer and bit count
SSSub: hcs \$list_change - request GIM to change DCW list
SSSub: hcs \$list_connect - request GIM to start device channel on DCW list
SSSub: hcs \$list_size - set size of GIM DCW list for device channel
SSSub: hcs \$make_ptr - return pointer to entrypoint given segment name and make known by given reference name (searches and makes known)
SSSub: hcs \$make_seg - create file branch given absolute path name and make known by given reference name, ring number
SSSub: hcs \$proc_info - return record quota, current use, and cumulative use for directory branch
SSSub: hcs \$quota_get - return record quota, current use, and cumulative use for directory branch
SSSub: hcs \$safety - request GIM to stop device channel and safety DCW list
SSSub: hcs \$set_alarm - request wakeup at given time in microseconds
SSSub: hcs \$set_bc - set bit count in file branch entry
SSSub: hcs \$set_consistsw - set consistency switch in branch entry
SSSub: hcs \$set_copysw - set copy switch in branch entry
SSSub: hcs \$set_relativ - set relate switch in branch entry
SSSub: hcs \$set_timer - request wakeup or fault after given CPU time interval
SSSub: hcs \$star_list - search a directory for entry name's matching template, return long information
SSSub: hcs \$status - return information from directory entry about a branch or link
SSSub: hcs \$status_long - return all information from directory entry about a branch or link
SSSub: hcs \$status_minf - return bit count and type for a branch or link
SSSub: hcs \$status_mips - return bit count and type for a segment
SSSub: hcs \$terminate_file - terminate all reference name's for segment from KST
SSSub: hcs \$terminate_name - terminate one reference name for segment from KST
SSSub: hcs \$terminate_seg - terminate all reference name's for a segment from KST
SSSub: hcs \$truncate_file - truncate file branch from a given offset
SSSub: hcs \$unassign_seg - truncate segment from a given offset
SSSub: hcs \$usage_values - request GIM to un-assign a device channel
SSSub: hcs \$wakeup - return cumulative page waits and CPU time used
AMSub: hcs \$wakeup - send inter-process wakeup to process with given id
SSSub: ioa - format a message and write it on "user_output"
SSSub: ioa \$ioa_stream - format a message and write it on given ioname
SSSub: ioa \$ioa_stream_nli - format a message and write it on given ioname, no newline
SSSub: ioa \$nli - format a message and write it on "user_output", no newline
SSSub: ioa \$rs - format a message and return character string
SSSub: ioa \$rsnl - format a message and return character string, no newline
SSSub: ioa \$abort - abort outstanding transaction's for an ioname
SSSub: ioa \$attach - associate one ioname with another ioname, a mode, and an outer module
SSSub: ioa \$change_mode - change mode of attachment of an ioname
SSSub: ioa \$detach - un-attach one ioname from another ioname
SSSub: ioa \$getdelim - return current real delimiters for ioname
SSSub: ioa \$getmode - return binary encoding of mode of attachment for ioname
SSSub: ioa \$getsize - return current element size for an ioname
SSSub: ioa \$lowait - wait for transaction's on workspace asynchronous ioname
SSSub: ioa \$order - issue request to outer module associated with an ioname
SSSub: ioa \$read - read from an ioname
SSSub: ioa \$read_ptr - read from ioname "user_input"
SSSub: ioa \$ready - set real synchronization mode for an ioname

```

SSSSub: ios $resetread - delete unused read-ahead collected for an ioname
SSSSub: ios $resetwrite - delete unused writes-behind collected for an ioname
SSSSub: ios $seek - set reference pointers associateX1 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 $stell - return value of reference pointer associateX4 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
AMISub: ipc $block - block process until given event
AMISub: ipc $chn_1 - return name of this validation level's event channel 1
AMISub: ipc $chn_2 - return name of this validation level's event channel 2
AMISub: ipc $create_ev_chn - create an event channel
AMISub: ipc $cutoff - inhibit event channel for reading
AMISub: ipc $decl_ev_call_chn - make event channel into event-call type
AMISub: ipc $decl_ev_wait_chn - make event channel into event-wait type (default)
AMISub: ipc $delete_ev_chn - delete an event channel
AMISub: ipc $drain_chn - drain event channel of any pending wakeup
AMISub: ipc $mask_ev_calls - request ipc $block not to interrogate event-call channels
AMISub: ipc $read_ev_chn - return event message from an event channel
AMISub: ipc $reconnect - enable event channel for reading (after cutoff)
AMISub: ipc $set_call_prior - request ipc $block to interrogate event-call channels before event-wait channels
AMISub: ipc $set_wait_prior - request ipc $block to interrogate event-wait channels before event-call channels
AMISub: ipc $unmask_ev_calls - request ipc $block to interrogate event-call channels
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_sparse_file_cut_line - return indices of current line of character file
SSSSub: parse_file_sparse_file_init - initialize parse_file with given segment name
SSSSub: parse_file_sparse_file_init_ptr - initialize parse_file with pointer to segment
SSSSub: parse_file_sparse_file_ptr - scan character file and return pointer to next break or symbol
SSSSub: parse_file_sparse_file_set_break - declare break characters for parse_file
SSSSub: parse_file_sparse_file_unset_break - undeclare break characters for parse_file
AMISub: random $exponential - return random number from exponential distribution
AMISub: random $normal - return random number from normal distribution
AMISub: random $normal_ant - return random number from antithetic normal distribution
AMISub: random $normal_seg - return array of random numbers from normal distribution
AMISub: random $random_get_seed - return current seed for random numbers
AMISub: random $normal_ant_seg - return array of random numbers from antithetic normal distribution
AMISub: random $random_set_seed - set seed for random numbers
AMISub: random $uniform - return random number from uniform distribution
AMISub: random $uniform_ant - return random number from antithetic uniform distribution
AMISub: random $uniform_ant_seg - return array of random numbers from antithetic uniform distribution
AMISub: random $uniform_seg - return array of random numbers from uniform distribution
AMISub: read_list - read in free-format variables from "user_input"
AMISub: reverse_index - scan character string from right to left for given character
SSSSub: reverse - revert a condition handler
AMISub: round - round a number to next multiple of 2**n
AMISub: round $rndshftb36 - round a number to multiple of 36, divide by 2**n
AMISub: round $rndshftb36 - round a number to multiple of 36, divide by 2**n
AMISub: round $rndshftb36 - round a number to multiple of 36, divide by 2**n
SSSSub: signal - signal the occurrence of a condition, call its handler
SSSSub: term - remove segment's linkage section, unshio linkage pointers to segment, and terminate segment
SSSSub: term $makeunknown - 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

```


SSSub: ti_\$findata - turn scratch segment into data segment
SSSub: ti_\$finobj - turn scratch segment into object segment
SSSub: ti_\$setseg - set up a scratch segment
SSSub: unique_bits_ - return unique bit string
SSSub: unique_chars_ - convert bit string into unique character string
AMSub: whoami_ - return user's name and project as given at login
AMSub: write_list_ - write variables on "user_output"

