TO: MSPM
FROM: G. F. Clancy
SUBJ: BG.17.01
DATE: 09/27/67

The attached revision to section BG.17.01 reflects recent changes to the Drum Utility Module. Major modifications are to the declaration of master hypersector and free storage map sectors.

Published: 09/27/67
(Supersedes: BG.17.01, 06/01/67;
BG.17.01, 10/28/66;
BG.17.02, 10/28/66)

Identification

Drum Utility Module (DUM)
S. H. Webber, G. F. Clancy

Purpose

The drum utility module is intended as a stand-alone system for use in circumstances when the Multics maintenance staff must examine or alter the contents of the firehose drum using trackset-sector addressing. The utility package acts as an interface between the operator typing requests at the 645 console and the drum.

Introduction

The operator has the following representative capabilities using the drum utility module:

(1) Examine small sections of the drum via the on-line typewriter (peek);

(2) Dump large blocks of the drum onto the printer (dump);

(3) Patch a particular word in a given record with a new value (patch);

(4) Save block(s) of records on tape for later reloading or printing (save);

(5) Reload block(s) of records from a previously written tape (restore);

(6) Overlay one drum record with the contents of another (copy);

(7) Create a free storage map on the drum (wrmap).

The module is written as a package with calls to "black box" I/O routines which can be replaced as Multics develops from a stepchild of GECOS to a stand-alone system. In particular, new routines to interface with the on-line typewriter, the drum, the printer, and magnetic tape will probably be substituted several times before the drum utility module stabilizes. The module will eventually reside on the Multics system tape from which it can be loaded.

In the discussion that follows, the term "record" applies to a 64 word block of data on the drum specified by a trackset-sector address. A "hyperrecord" is a number (power of 2) of contiguous sectors on the drum which are treated as a unit.

Addresses which are specified by an operator using the DUM are 18 bit integers. The DUM converts these addresses into trackset-sector address which the drum recognizes. The firehose drum then appears as 1 large linear device with 2^{18} "records" of data on it. The operator specifies the "record"(s) he desires: the DUM does the rest.

The drum utility module is a subset of the Device Utility Process (DUP) which includes utility modules for the MSU302 and MSU388 storage devices.

Device Partitioning

Each large capacity logical storage device of a Multics hardware system is capable of being partitioned into several contiguous regions. Only one region from any device will be used on any particular Multics configuration. The partitioning merely allows extra-Multics or multiple-Multics operation.

On the firehose drum there will be a maximum allowed number of 12 partitions. (Note: In what follows the term "partition" will be used to indicate an element of the partition of the entire drum. Thus one might say "define partition 1" and mean define the limits of the first subregion of the drum.) This does not mean that there cannot be less, say 1 for the entire drum, but rather that as many as 12 different, disjoint subregions, of different sizes possibly, may be defined.

Associated with each partition are certain parameters which hold throughout that partition (i.e. throughout that subregion). These are:

1. hyperrecord size - that is, the number of records per hyperrecord (rphr)

2. master hyperrecord - the location (relative hyperrecord address) of the master hyperrecord for this partition. The master hyperrecord is used in handling free storage areas within the partition. (master)

3. reclo - the relative hyperrecord address which is the lower bound for this partition (reclo)

4. abreclo - the absolute record address which is the lower bound for this partition. Note that (abreclo)

abreclo = reclo+rphr+c

where

c < rphr.

5. rechi - the relative hyperrecord address which is the upper bound for this partition. (rechi)

6. abrechi - the absolute record address which is the upper bound for this partition. (abrechi)

These quantities are stored in the configuration record (one per device) of the device. In addition the following information is kept within the configuration record for each partition:

1) The current length and maximum length of the root directory (if the root directory is kept on this device).

2) A switch, psw, which indicates whether (ON) or not (OFF) the current partition is defined.

3) A switch, initsw, which indicates whether (ON) or not (OFF) the current partition has been initialized (free storage maps written, etc.).

In addition the configuration record contains a count of how many records the current device contains as well as a count of the number of partitions currently defined.

A declaration for the configuration record is found in Appendix 1.

Command Format

A command line may consist of several commands separated by semicolons (";"). Numbers specified as arguments to commands are assumed to be octal unless immediately preceded by a "d" to indicate a decimal number or by a "b" to indicate a binary number. Abbreviations for each command can be

found in Appendix 2 along with the calling sequences. Optional arguments are enclosed between minus signs ("-") and may be omitted or specified by a "*". In either case each assumes the indicated default value.

DUM Commands

Appendix 2 contains a list of all the commands which are available under the DUM. The accepted abbreviation and calling sequence are also given for quick reference. This section describes in more detail the command, the appropriate call, the default values for optional arguments, and possible error conditions. Appendix 3 lists the error codes and their related meanings as returned by the DUM.

Absolute

This command sets the DUM into absolute record addressing mode. This means that the following static variables are reset as indicated:

rphr = 1

reclo = 0

rechi = no_records - 1

part = 0

copy hrecord1 hrecord2

This command copies hyperrecord hrecord1 into hrecord2 of the currently defined partition. Hyperrecord hrecord1 is not changed.

Restriction

Both hyperrecords specified must be within the bounds of the currently selected partition.

define part rphr abreclo abrechi

This command establishes a drum partition and hyperrecord size to be used by future commands or subroutine calls. rphr becomes the new hyperrecord size (number of 64 word records per hyperrecord). abreclo defines

the zeroth absolute record address for the partition while abrechi specifies the last absolute record address for the partition. Figure 1 illustrates a typical partition.

The partition defined is given by part and consists of the contiguous records from abreclo to abrechi.

The command "define" generates a call to the procedure define_device which checks the arguments against other defined partitions and in general against illegal specifications. In particular two partitions may not overlap and no partition may include the configuration records.

Define_device not only establishes one or more partitions but also initializes the root_directory (to zeroes) if root_sw is ON, initializes the free storage maps, and sets up the master hyperrecord. These are done with calls to the following procedures:

wrmap

make_root

The argument passed to define_device is a structure (whose declaration may be found in appendix 1) which is set up either by the "define" command or by the file system initialization procedures. Within this structure is the above mentioned "root_sw" which indicates whether or not the root directory is to reside on this device. If "root_sw" is ON, "make_root" is called to reserve hyperrecords for the root directory, zero these out, and set up the current and maximum length of the root-directory in the configuration record.

Assigned Values

reclo = first absolute hyperrecord address which is greater than or equal to abreclo and is zero modulo rphr.

rechi = absolute hyperrecord address of the last complete hyperrecord with an address less than or equal to abrechi, i.e.

$$(rechi + 1) * rphr \leq abrechi$$

and rechi is zero modulo rphr.

Note: reclo and rechi are absolute hyperrecord addresses whereas abreclo and abrechi are absolute record addresses.

Default values

The following default relative hyperrecord addresses apply:

master hyperrecord; 0

root_directory hyperrecords; 1 thru ml/rphr + 1

where "ml" is the number of 64 word blocks reserved for the root_directory. "ml" must be at least 16 and must be a multiple of 16.

Restrictions

1) part must be between 1 and 12

2) rphr must be a power of 2 and within the bounds of 2 to 2^8

3) the configuration record (default sector 0) must not be between abreclo and abrechi.

4) rechi - reclo $\geq$ 512 (to guarantee enough free storage map addresses).

5) the partition defined by abreclo and abrechi must be consistent with other defined partitions (i.e. no overlap is allowed).

relative hyperrecord address	absolute hyperrecord address	absolute record address
0	reclo	reclo * rphr
1	reclo + 1	(reclo + 1) * rphr
2	reclo + 2	(reclo + 2) * rphr
.	.	.
.	.	.
.	.	.
rechi - reclo	rechi	rechi * rphr

Figure 1

dump hrecord -count-

This command will dump hyperrecords hrecord through hrecord +count-1 on the on-line printer.

Default values

1) count = 1

Restrictions

1) hrecord must be between 0 and rechi-reclo
2) count must be between 1 and rechi-hrecord + 1

dumpc hrecord -count-

dumpm hrecord -count-

These alternate dump commands give the operator certain freedom in specifying the format of the dump. "dumpc" will dump the specified records in free storage map ("chain") format. "dumpm" will dump the specified records in master hyperrecord format.

These 3 alternate dump commands have the same default values and restriction as the "dump" command has.

external static

This command types the following static variables on the on-line console:

rphr
reclo
rechi
master
config
part

make-root part

This command sets up a call to the procedure "make_root" (much as define_device does) which does the following:

1) create (reserve storage for) the root_directory

2) zero out the root_directory

3) stores the following in the configuration sector for partition part

   a) drum addresses of the root directory
   b) maximum length (in 1024 word blocks) of the root directory
   c) current length (in 64 word blocks) of the root directory --- set to 0.

Restrictions

1) If "make_root" is called for a partition which has initsw ON, an error return is taken ("define" does not turn initsw ON until after "make_root" is called).

2) part must be between 1 and 12.

select part

This command selects the partition part from among the several partitions currently defined in the configuration sector. The static variables are reset to reflect the change.

Restrictions

part must be between 1 and 12

patch hrecord wordno patchword

This command allows the user to patch specified words of specified records. The wordno word of relative hyperrecord hrecord will be replaced with the binary equivalent of patchword.

Restrictions

1) patchword may consist of only octal digits

2) wordno must be less than or equal to rphr*64-1

3) hrecord must be within the bounds of the partition selected.

peek hrecord -wordno- -count-

This command types on the on-line console count words of relative hyperrecord hrecord starting at word wordno. (There are 64*rphr words in a hyperrecord numbered 0 through 64*rphr -1.)

Default values

1) wordno = 0
2) count = 64*rphr - wordno

Restrictions

1) hrecord must be within the bounds of the selected partition.

Note: if wordno is greater than 64*rphr -1 then wordno is set to 64*rphr -1. If wordno+count -1 is greater than 64*rphr -1, count is set to 64*rphr - wordno.

quit

The normal means of terminating the DUM is with the "quit" command. This returns control from the main procedure of the DUM, drmutil.

reserve record

This command adds the absolute hyperrecord address record to the reserved list with the appropriate error return if record is already on the reserved list. In addition the error return is taken if too many records are already reserved. "no_reserved" is incremented by 1.

Restrictions

1) record must not be on the reserved list.

2) If there are already 16 records on the reserved list, no more may be reserved.

restore filename -record- -count-

restore to printer filename -record- -count-

These commands may be executed only if in "absolute mode". i.e. only if the following values are current:

```
reclo  0
rechi  no_records-1
rphr   1
```

restore searches the tape file filename for records record through record + count -1 and those found are rewritten on the drum. If the record range specified by record and count is not a subset of the range of file filename a message will be typed on the on-line console indicating those records which may be restored. The operator may then abort the command or load designated records. restore to printer does the same as restore except the I/O stream is channeled to the printer instead of the drum. (The drum is not altered.)

Default values

1) record = 0
2) count = no_records - record

Restrictions

1) record must be greater than or equal to 0

2) count must be at least 1 and less than or equal to no_records - record

3) the DUM must be in absolute mode

save filename -record- -count-

This command may be executed only if in "absolute mode".

This command will save records record through record+count -1 on magnetic tape. The tape reel will be specified by filename.

Default values

1) record = 0
2) count = no_records-record

Restrictions

1) record must be greater than or equal to 0

2) count must be at least 1 yet less than or equal to no_records-record

3) the DUM must be in absolute mode

set_static name1 value1 ... -namen- -valuen-

This command allows manipulation of the static variables used by the DUM. To change one of these variables, the code name must be given followed by the new value. The following is a list of variables which may be changed in this manner (along with accepted abbreviation).

reclo	rl
rechi	rh
rphr	rp
part	p
no_records	nr
master	ma
config	cf

E.g. set_static rphr 4 reclo 0 p 0

status -part-

This command allows the operator to find how each of the defined partitions is indeed defined. The following values are types out on the on-line console for partition part.

master_hr
current length (of root directory)
maximum length (of root directory)
psw - processed switch
initsw - initialized switch
abreclo
abrechi
records per hyperrecord

If part is 0 all partitions are revealed.

Restrictions

1) part must be between 0 and 12

Default values

1) part = 0

test1 -arg1- -arg2- ... -argn-

test2 -arg1- -arg2- ... -argn-

Test1 and test2 are two commands which allow a user of the DUM to create his own command. The DUM merely generates a call out to the procedure test1 (or test2) and passes as an argument the character array consisting of whatever arguments were typed in the command line. It is up to test1 (or test2) to convert and/or interpret the character strings passed to it.

undefine part

This command undoes the work of "define" by setting "psw" OFF for partition part in the configuration record. This command must be issued when it is desired to remove the partition part from the configuration record.

Note: "define" will not rewrite a partition (redefine it) if the partition is currently defined.

unreserve record

This command removes the absolute hyperrecord address record from the reserved list with the appropriate error return if record is not on the reserved list. no_reserved is decremented by 1.

Restrictions

1) record must be on the reserved list

write_map -no_chains-

This command issues a call to the procedure "wrmap" which creates free storage maps for the partition currently selected. The free storage map format consists of several (2-16) chains of free storage "offrecords". These offrecords contain free storage addresses and are threaded forward and backward and to the master hyperrecord. The master hyperrecord contains the following variables:

1) number of chains
2) current deposit chain
3) current withdrawal chain
4) records per hyperrecord

In addition for each chain the following variables are kept:

1) first hyperrecord
2) number of addresses in first
3) last hyperrecord
4) number of addresses in last

Declarations for the master hyperrecord and for "offrecords" are given in Appendix 1.

Default value

no_chains = 2

Restrictions

1) nochains must be between 2 and 16

2) records per hyperrecord must be greater than 2 and a power of 2.

APPENDIX I

Declarations used by the DUM

1) The master hyperrecord

```
dcl 1 master based (masptr),
    2 miow,
      3 ok bit (9),
      3 pending bit (9),
      3 error bit (18),
    2 diow (16),
      3 ok bit (9),
      3 pending bit (9),
      3 error bit (18),
    2 wiow (16),
      3 ok bit (9),
      3 pending bit (9),
      3 error bit (18),
    2 no_sectors fixed bin (17),
    2 catastrophe_flag bit (1),
    2 no_chains fixed bin (17),
    2 max_chains fixed bin (17),
    2 cur_depo_chain fixed bin (17),
    2 cur_withdr_chain fixed bin (17),
    2 diolock bit (36),
    2 wiolock bit (36),
    2 dbflock bit (36),
    2 wbflock bit (36),
    2 dend (16) fixed bin (18),
    2 dndx (16) fixed bin (17),
    2 wend (16) fixed bin (18),
    2 wndx (16) fixed bin (17),
    2 next_chain_lock bit (36),
    2 next_chain (16) fixed bin (17);
```

2) The offrecord (chain record)

```
dcl 1 offsect based (offptr),
    2 checksum bit (36),
    2 seq_num bit (36),
    2 chain_num bit (36),
    2 back_addr bit (18),
    2 forward_addr bit (18),
    2 addr (248),
      3 ess bit (18);
```

APPENDIX I (continued)

3) The configuration record

```
dcl 1 conf based (bfp),
    2 part (12),
      3 master_hs bit (18),
      3 sphs bit (18),
      3 psw bit (1),
      3 initsw bit (1),
      3 cur_root_length bit (13),
      3 max_root_length bit (9),
      3 unused bit (12),
      3 root_file_map bit (180),
      3 absectlo bit (18),
      3 absecthi bit (18),
      3 sectlo bit (18),
      3 secthi bit (18),
      3 dummy bit (18),
      3 no_reserved bit (18),
      3 reserved (16) bit (18);
```

APPENDIX II

The calls (and abbreviations) available with the DUM.

ab

absolute

---

c

copy record_1 record_2

calls: check$sect, dup$copy

---

df

define part rphr abreclo abrechi

calls: define

---

d

dump hrecord -count-

calls: check$sect, check$count, check$mode

---

dc

dumpc (same as dump)

dm

dumpm (same as dump)

---

es

external_static

---

mr

make_root part

calls: check$part, make_root

---

sl

select part

calls: check$part, select

---

pt

patch hrecord wordno patchword

calls: check$sect, check$word, dup$patch

---

pk

peek hrecord -wordno- -count-

calls: check$sect, check$word, check$count, dup$peek

---

q

quit

---

rs

reserve record

calls: reserve$in, check$sect

---

r

restore filename -record- -count-

calls: restore

---

rp

restore_to_printer filename -record- -count-

---

s

save filename -record- -count-

calls: save

---

ss

set_static name1 value1 ... -namei- -valuei-

calls: set_static

---

st

status -part-

calls: dup_status, check$part, check$mode

---

t1

test1 -arg1- -arg2- ... -argn-

calls: test1

---

t2

test2 -arg1- -arg2- ... -argn-

calls: test2

---

ud

undefine part

calls: check$part, undefine

---

ur

unreserve record

calls: reserve$out, check$sect

---

w

write_map -no chains-

calls: wrmap

APPENDIX III

ERROR CODES

OCTAL VALUE	COMMAND	MEANING
000000	ALL	ILLEGAL PARTITION -- "PART" IS NOT BETWEEN 1 and 12
000001	ALL	DRUM I/O ERROR
000002	ALL	ILLEGAL INPUT CHARACTER STRING
000003	ALL	"PART" SPECIFIES A PARTITION WHICH IS EITHER NOT INITIALIZED OR NOT DEFINED
000004	PATCH	"PATCHWORD" HAS NON-OCTAL CHARACTERS IN IT
	SAVE	EITHER "RECORD" OR "COUNT" IS NOT ACCEPTABLE
	SET_DEVICE	NO SUCH DEVICE
	DEFINE_DEVICE DEFINE WRMAP	"RECORDS PER HYPERRECORD" IS NOT A POWER OF 2 OR "RECORDS PER HYPERRECORD" LESS THAN OR EQUAL TO 1
	RESTORE	RANGE EXCEEDS FILE BOUNDS
	STATUS	"MODE" NOT EQUAL TO 0 OR 1
	MAKE_ROOT	"INITSW" IS ON FOR PARTITION "PART"
000005	SAVE	NOT IN ABSOLUTE MODE
	WRMAP	"NO_CHAINS" LESS THAN 1 OR GREATER THAN 16
	DEFINE_DEVICE DEFINE	"RECHI" - "RECLO" LESS THAN OR EQUAL TO 512
	RESTORE	NOT IN ABSOLUTE MODE
000006	DEFINE_DEVICE DEFINE	INCONSISTENT WITH OTHER PARTITIONS

OCTAL VALUE	COMMAND	MEANING
000007	DEFINE_DEVICE DEFINE	CONFIGURATION RECORD WITHIN THE PARTITION BOUNDS
000010	DEFINE_DEVICE DEFINE	ASKED TO DEFINE LESS THAN 1 OR MORE THAN 12 PARTITIONS
000012	ALL	"RECORD" IS OUT OF BOUNDS
000013	ALL	"WORDNO" IS NOT ACCEPTABLE
000014	ALL	"MODE" IS NOT 0 OR 1
000015	ALL	"PART" IS NOT BETWEEN 1 AND 12
000016	ALL	"COUNT" IS NOT ACCEPTABLE
000017	ALL	NOT ENOUGH ARGUMENTS
000020	RESERVE	"RECORD" ALREADY RESERVED
000021	RESERVE	NO MORE RECORDS CAN BE RESERVED
000022	RESERVE	"RECORD" IS NOT ON THE RESERVED LIST
000023	RESERVE	"RECORD" NOT WITHIN PARTITION BOUNDS
000025	DRUM_IO	DRUM DID NOT CONNECT
000026	DRUM_IO	DATA FAULT
000027	DRUM_IO	CONTROL FAULT