Identification

Drum Utility Module
S. Kidd, 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 and/or patch the contents of the firehose drum using absolute track addressing. The utility package acts as a thin interface between the operator (system programmer) typing requests at the 645 console, and the drum.

Introduction

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

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

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

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

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

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

(6) Overlay one drum track 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 substitued several times before the drum utility module stabilizes. The module will eventually reside on the Multics system tape from which it can be loaded.

Usage

The following are the specific commands of the drum utility package. The first word of each line is the command name and all following words except "filename" are to be numbers with the indicated meaning. Numbers will be assumed to be octal unless immediately preceded by a "d" to indicate a decimal value. The abbreviation of each command is given beneath the full name.

Bracketed groups of arguments can be omitted. If they are left out they will be given default values.

(1) peek trackset sector [word1 [word2]]

pk

Type on the online console the contents of the drum sector defined by trackset - sector from word1 to word2. If word2 is omitted, it will be given a default value of 64. If word1 is also omitted, its value will be 0.

(2) patch trackset sector j "word"

pt

Replace the jth word of trackset-sector with the value "word".

(3) dump trackset [sector1 [sector2]]

Print on the high speed printer the contents of sectors sector1 to sector2 of trackset. Default value of sector1 is 0, of sector2 the last sector in the trackset.

(4) save filename [trackset1 [trackset2]]

s

Dump the contents of all of trackset1 to trackset2 on tape with file name "filename" (an alphameric identifier). If only trackset1 is given, all tracksets to the end of the drum will be saved. If neither trackset1 nor trackset2 is given, the entire drum will be saved. Each trackset will be written with an identifying header so that sections of the drum can be reloaded selectively.

(5) restore filename [trackset1 trackset2...]

r

The tape file filename will be searched for records containing data saved from trackset1 ... tracksetn, and those will be rewritten on the drum from the tape. A message will be typed on the console for any tracksets not found in filename.

If no tracksets are specified all the records in filename will be written onto those drum sectors specified in the record leaders, i.e. returned to exactly those locations from which they were saved.

(6) wrmap trackset sector chainlength

w

Create a free storage map on the drum with chainlength elements and with its origin at trackset-sector.

(7) copy trackset1 sector1 trackset2 sector2

c

Read the 64 word block at trackset1-sector1 and write it into trackset2-sector2.

Since sector addresses are often represented as 18-bit integers rather than by the two components "trackset" and "sector number" the pair of arguments "trackset sector" can be combined to a single octal number by preceding that number with "=". For example, the sector address "3 17" could be expressed "=617". Multiple requests may be put on a line when they are separated by ";". An interrupt during output will kill the rest of the output.