



DECUS

PROGRAM LIBRARY

DECUS NO.	8-270
TITLE	DISK-DECTAPE UTILITY PROGRAM
AUTHOR	Garth Peterson
COMPANY	South Dakota School of Mines and Technology Rapid City, South Dakota
DATE	March 3, 1970
SOURCE LANGUAGE	PAL D

DECEMBER



DISK-DECTAPE UTILITY PROGRAM

DECUS Program Library Write-up

DECUS No. 8-270

INTRODUCTION

The Disc-DEctape Utility Program (DDU) performs a variety of utility functions for PDP-8 users who have both DF32 disc and DEctape. To begin with, DDU can save disc images on DEctape and later recover them. This allows one or more users at an installation to build and operate several different disc systems. A user with only one disc but with DEctape will find it worthwhile to save a basic disc monitor with only system programs; this monitor can then be a foundation for several different disc monitors, each of which has the ASCII and user files pertinent to a particular programming job. DDU can bootstrap out to a disc or DEctape monitor system (DEC-08-SDAB) or to a DEctape library system; thus the system to which DDU exits can be different from the system which brought it into core. DDU also supports a "disc library system". This is a short executive routine which resides in the first 128 words of disc 0 and which swaps programs between field 0 and disc 0. It can be useful only when magnetic tape is available for saving and recovering disc monitors, which are needed for assemblies. Under the disc library system, however, user-written programs can store data on disc 0, in places not used for program storage, while addressing the disc as simply as one chooses and transferring data in blocks of up to 4096 words instead of 128. Additional functions of DDU will be described below.

GENERAL CHARACTERISTICS

At the beginning of execution DDU types first its name and then a carriage return, line feed, and asterisk to indicate readiness for a command. Thereafter the program starts a new line and types an asterisk to ask for a new command whenever a function has been aborted or completed, except for bootstraps completed. In general a command consists of one line, for which the operator must enter the initial character and device identification characters, with the program filling in intervening characters automatically. Commands for functions which alter the contents of a disc or DEctape must be finished by a carriage return from the keyboard. In the descriptions below characters which must be entered by the user are underlined. All graphic characters entered while keyboard input is sought--and not before--will be echoed on the teleprinter. Keyboard input at other times will be ignored except that keyboard input will foul things up while a binary tape is supposed to be read on the teletype.

If a rubout or invalid character is entered when the program is waiting for keyboard input, two question marks are typed out and the function is aborted. Disc or DEctape errors will abort a function and cause "? DC ERR" or "? DT ERR" to be typed.

At the start of the program, even before typing its name, DDU checks the system head to see if it is from a disc or DECTape system and then overlays it with a binary loader and a RIM loader, making it necessary to use a bootstrap in DDU to return to the system. The standard version of the program supplies a RIM loader for the high-speed reader. The binary loader will be described under the "RCVY" function.

DISC IMAGE FUNCTIONS

"*SAVE DC d, DT u PART p c/r"

"*GET DC d, DT u PART p c/r"

These functions save or get up to five separate disc images on one DECTape. A disc select number from 0 to 3 is entered at "d", "u" is a DECTape unit number from 1 to 8, (Unit 8 may also be specified by a 0.) and "p" is a digit from 0 to 4. Part 0 of the DECTape starts at block 0010 (octal). Disc images are stored consecutively with 128 disc words in each DECTape block and with 256 blocks per disc image.

DECTAPE FUNCTIONS

"*MSG, DT u BLK b"

After the above command the program starts a new line and permits the operator to enter a single line of not more than 72 characters of text. When the message is terminated by a carriage return, the text is stored in block b of the DECTape on unit u. The block number cannot be greater than 7. While the text is being entered, a rubout will remove the last character in the text buffer and will be echoed as a backward slash except that if the buffer is empty a rubout has no effect and will not be echoed. A back arrow will ordinarily remove all of the current contents of the text buffer and will allow the operator to start the message over on a new line, but a back arrow entered while the text buffer is empty will abort the function. It is not an error to enter a carriage return while the buffer is empty; this stores a blank message on the DECTape. Messages stored by this function may be examined by the next function. Furthermore a message in block 0 will be typed out by instructions stored along with it if the DECTape is bootstrapped by mistake. A RIM loader will also be present.

"*EXAM MSG, DT u BLK b"

This function reads a message of the above type from a DECTape and outputs it on the teletype.

"*ZAP DT u c/r"

The specified DECTape is cleared to all zeroes. This is accomplished by writing 129 words with the value 7777 (octal) into each block while the DECTape is moving backwards.

"*COPY DT u ONTO v c/r"

The DECTape on unit u is copied onto the tape on unit v. If u = v, the program will halt between read and write operations to allow tapes to be changed on a single handler. With either one handler or two, this is not an efficient copying routine since DDU's data buffer holds only 16 blocks, and there is no automatic rewind.

MISCELLANEOUS FUNCTIONS

DDU is a single-field program in that all of its instructions, plus its data buffer, lie in one memory field. Nevertheless, DDU can run in any field and can copy itself back and forth between field 0 and one of the extended memory fields. This extended memory field will be referred to below as the "alternate" field. The alternate field is normally field 1, but if DDU is loaded into some field other than 0 or 1, it will redefine the alternate field as the field it is in as soon as it is started.

"*FLIP TO Fn"

The program copies itself into field n (an octal digit) and then asks for a new command, continuing its execution in the new field. If DDU is in field 0, "n" is the alternate field, and vice versa. This command and the next one are invalid on a 4K PDP-8.

"*RCVY IN Fa"

This command stores a recovery routine consisting of a DECTape bootstrap (starting address a7600), a paper tape binary loader (a7777), and a disc bootstrap (a7756) in the last page of the alternate field, a (an octal digit). The bootstraps are not affected by the starting contents of the data field register; in the event of disc or DECTape error, both of them stop on the halt instruction of the binary loader, making it necessary to reload the starting address into the program counter. The binary loader loads into the starting data field until a field setting character on the input tape intervenes; when the START key is pressed, switch 0 must be on for teletype input and off for the high speed reader. This binary loader must be started on leader characters and not on tape ahead of the leader. Embedded diagnostics are allowed, field setting characters are not added to the calculated checksum, and there is no protection against self-overlay. This is the same binary loader which is deposited in the last page of field 0 when DDU is started. It uses the disc and DECTape data break locations (7750, 7751, 7754, 7755) but only for temporary storage.

"RCVY" may be saved as a separate monitor system program by the monitor command ".SAVE RCVY! a7600; 0 c/r". On an 8K machine this provides a quick way to restore a binary loader in field 1 after an 8K software program has been run. It also enables the user to bootstrap a DECTape when the disc monitor is too full to hold DDU.

It is appropriate to mention here that the bootstraps and loader described under the "RCVY" function are also deposited in every whole page of DDU's data buffer (3600 to 7577 within the current field) whenever carriage return, line feed, asterisk are typed out.

"*DC BTS"

The disc system, either monitor or disc library, is bootstrapped.

"*TAPE BTS"

DECTape unit 8, which should have either a monitor or library system, is bootstrapped.

"*control-C"

If the C key is pressed with the control shift key down, a disc or DECTape bootstrap function will be performed, including typing of the command, depending on whether DDU found a disc or DECTape system head in the last page of field 0 at the start. Control-C is valid only as the first character of a command.

"*carriage return"

A carriage return starts a new line and causes a new asterisk to be typed.

"*."

A period will cause DDU to halt at 0201 in the current field. Program execution will resume when the START or CONTINUE key is pressed.

DISC LIBRARY FUNCTIONS

Under the disc library system disc 0 is logically divided into eight "sectors" of 4096 words each, numbered from 0 to 7. Let s denote the sector digit. Then each sector can store one field 0 program, with disc addresses 0s0200 to 0s7777 corresponding to core addresses 00000 to 07577. The library head resides in disc addresses 000001 to 000177 and in core addresses 07600 to 07776. The first word of the disc is omitted for bootstrap compatibility with the disc monitor system. Any disc location not used by a program or by the library head is available for data storage.

When the library head is bootstrapped into core, it will type carriage return, line feed, per cent sign to indicate readiness for a command, and there will be a RIM loader at 7756. In the following disc library commands "s" denotes a sector digit:

"%sW"

The program in field 0 is written into disc sector s, and the library head is ready for a new command.

"%sR"

The program in disc sector s is read into field 0, but control remains in the library head.

"%sG"

The library head reads a program into core and then branches to 0200. If 0200 is not the starting address, the "R" command and a manual start must be used.

"%G"

This causes a branch to 0200 with no disc operations.

If the library head detects an invalid command or a disc error, it types a question mark and asks for a new command. Multiple sector digits before the terminal letter of a command are legal; the disc operation is performed on the sector of disc 0 specified by the last digit. A non-existent disc condition is not detected as an error and causes an indefinite loop.

The following DDU commands support the disc library system:

"*LIB HD (DC) c/r"

A library head is written on disc 0.

"*WR FO ONTO DC SCTR s c/r"

The program in field 0 (maybe DDU or maybe not) is written onto disc sector s.

"*PULL INTO FO, DC SCTR s c/r"

A program is read into field 0 from the disc. This command is valid only if DDU is in its alternate field, and thus is always invalid on a 4K PDP-8.

"*BIN TO DC SCTR s, IN: d c/r"

A binary paper tape is loaded into a disc sector. The device, "d", is indicated by "R" for the high speed reader and by "T" for the teletype. If the high-speed reader is used, position the tape in the reader before entering the command. When the tape has been read, three error messages are possible: "CKSM ERR", "ADDR ERR", "FLD ERR". A checksum error does not prevent any tape data from being written on the disc. An address error occurs when one or more core words on the input tape fail to go on the disc because their addresses are in the last page of field 0. Core words for an extended memory field cause a field error message and also fail to load. Embedded diagnostics are ignored. Binary tape data not preceded by a field setting are presumed to have field 0 addresses. Disc locations not loaded by a binary tape are left undisturbed to allow this function to act as an initial loader or as a patch tape loader. It is permissible to start the paper tape ahead of the actual leader. To minimize stops by the high speed reader it is desirable for the input binary tape to change its loading address only infrequently between the ranges 0 to 3777 and 4000 to 7577.

"*leader-trailer"

Leader-trailer codes entered from the teletype before the first character of a new command will be ignored.

CONCLUSION

The standard binary tape of DDU contains RIM loaders for the high speed reader where RIM loaders are referred to above. A binary tape with teletype RIM loaders patched in is available. DDU is also available in a RIM format tape of the standard version, and this RIM tape can also be read by a binary loader. A binary tape of DDU contains no field setting characters.

The appropriate monitor save command for DDU is ".SAVE DDU! 0-3377; 200 c/r". However, 201 may be used as a static system starting address (a halt before execution).

If the user's PDP-8 has 12K core or more, and if an alternate field higher than field 1 is desired, DDU should be loaded into the desired alternate field. It can then be saved by a monitor as a program for that field, or it can be started, flipped, used to bootstrap, and saved as a field 0 program.

The program listing consists of three parts. The first part is the printer output of a card-to-paper-tape conversion program and shows the source program with comments. Lines which begin with a dollar sign are control cards which do not punch tape. The second and third parts are the symbol table and the third-pass assembly listing with comments omitted. Because the printer used had only 48 characters, certain PDP-8 characters are represented by superimposed characters, as follows:

Value

Superposition

"
[
]
form feed
%
?
@

:

Q '
(-
) -
FO
P /
Q .
AT
I =
* .

