DECUS
PROGRAM LIBRARY

DECUS NO.	8-204
TITLE	PATCH — A PDP-8 Binary Paper Tape Patch Program
AUTHOR	C. A. McComas
COMPANY	Digital Equipment Corporation Maynard, Massachusetts
DATE	April 18, 1969
SOURCE LANGUAGE	PAL III

PATCH - A PDP-8 BINARY PAPER TAPE PATCH PROGRAM

DECUS Program Library Write-up DECUS No. 8-204

Ø. Introduction

Purpose of the Program:

PATCH provides a simple, convenient means for making changes to PDP-8 binary format paper tapes, and for creating short binary tapes. Single binary tapes may be patched or merely copied. Several tapes may be combined, with or without changes being made. Additions to tapes may be created. Whole pages of code may be moved or deleted. In certain cases binary words may be inserted within existing binary data. Field expressions may be inserted, changed or deleted.

Changes and creations are made via the teletype keyboard using simple commands. Data is typed in octal.

Necessary Equipment:

PATCH will run on any PDP-5, 8-family computer with 4K memory. It allows either high or low speed input-output.

1. General Operation

Loading and Starting:

The Starting Address of PATCH is Ø20Ø/. PATCH will run in any field of memory.

Load PATCH using the Binary Loader. PATCH occupies locations ØØØ1-4377 of whatever field it is loaded into.

Initialization:

When PATCH is started at Ø2ØØ it will begin by typing the question "HIGH SPEED READER?" If the user responds with "Y", PATCH will echo "YES" and initialize the high speed reader. If the response is "N", PATCH will echo "NO" and initialize the ASR reader. Any other responses are illegal.

PATCH will next type "HIGH SPEED PUNCH?" Answers similar to those for the reader will produce corresponding results.

Next PATCH types "HOW MANY INPUT TAPES?" To this query the user must respond with an octal number followed by a carriage return. This number will be 1 if the user desires only to make changes to or copy a single binary tape. However, it may be more than 1 (2-7777) if the user wishes to combine several binary tapes into one, with or without patching. The number of input tapes may also be Ø. This option allows the user to create binary format tapes directly from octal data.

Note: if the user is using a high speed I/O device, he should load his input tape into the reader and turn his high speed I/O devices on before responding to the above question. If PATCH is unable to read a tape when it should, the error diagnostic "ILLEGAL TAPE" occurs. If PATCH is unable to punch because the high speed punch is off, the program will hang until the punch is turned on.

Responses:

The user may respond to any question typed by PATCH before the question is completed. If this is done, the question typing will stop and PATCH will immediately begin processing the user's response.

Error Diagnostics:

Generally speaking all PATCH error messages are self-explanatory. The most common error which can be made is an illegal user response to a PATCH question. The error message for this is simply

a "?" followed by a re-typing of the question or a return to an appropriate place in the program where the user may try again.

If there is anything wrong with a paper tape which the user is trying to input, PATCH will generate the diagnostic "ILLEGAL TAPE."

The few other possible error messages are mentioned in the text below where they are most appropriate.

The Output Format:

PATCH outputs standard PDP-8 binary format paper tapes suitable for loading by the Binary Loader. No matter how many input tapes are specified, PATCH produces a single output tape. This binary output tape is correctly checksummed and preceded and followed with leader/trailer code 2ØØ. PATCH will not output a double origin such as is frequently done by PAL at the beginning of a tape. Only the significant origin is used.

Termination:

Before ending any job PATCH always types the question "IS ADDITIONAL CODING DESIRED?" The user may answer this question with a "Y" for "yes" as many times as he pleases. PATCH will not terminate the job (and the output tape) until the user has responded to this question with an "N" for "no."

Restarting:

Whenever PATCH has reached the end of a job or when the "ILLEGAL TAPE" error has occurred, PATCH may be restarted to run a new job by merely pressing the Continue switch.

At any time PATCH may be stopped and restarted at Ø2ØØ by using the Console switches.

Low Speed I/O:

When using PATCH with low speed (ASR) I/O devices the following special instructions apply:

Whenever PATCH requires to pass from its teletype dialog phase to its input-output phase using low speed I/O equipment, or vice-versa, the program will halt. At this time the user should appropriately turn on or off whichever low speed I/O devices he is using, then press the Continue switch.

Patch Operation:

PATCH uses a single 2ØØ word buffer. This buffer represents a

single core page of code. Thus the buffer can never contain data for different core pages. Essentially PATCH operates by reading one page of code at a time into its buffer and then punching it out, after patching if patching is requested. Reading into the buffer stops each time an end-of-page condition is sensed. This occurs whenever the page fills up or whenever a reorigin or a field expression is encountered.

In the remainder of this document explanations are given for several different ways of using PATCH. The special options are all explained in the last section of the document.

Note: In all examples in this document user responses are underlined.

Throughout the PATCH program and this document "↑P" stands for the control-P character. "↵" stands for carriage return and "↓" for line feed.

2. To Copy a Single Tape or Combine Several Tapes

Copying a Single Tape:

To simply copy a tape with no patching proceed as shown below.

HOW MANY INPUT TAPES? 1↓
TYPE ADDRESS OF FIRST PATCH: ↓

Patch will now begin reading the input tape and punching it page-by-page when the last word of input data (except the checksum) has been punched PATCH will continue as follows:

IS ADDITIONAL CODING DESIRED? NO↓

At this point PATCH will punch the checksum and trailer code, and then halt.

Combining Several Tapes:

An example is given below showing how to combine several (e.g., 3) binary tapes into one.

HOW MANY INPUT TAPES? 3↓
TYPE ADDRESS OF FIRST PATCH: ↓

PATCH now copies the first input tape without, however, punching checksum or trailer.

LOAD NEXT TAPE.

At this point PATCH halts. The user should press the Continue switch when he is ready with his second tape.

TYPE ADDRESS OF FIRST PATCH: ↓

PATCH now copies the second tape, appending it directly to the output of the first tape, with no separation except for a reorigin.

LOAD NEXT TAPE.

Patch halts. When ready the user presses Continue.

TYPE ADDRESS OF FIRST PATCH: ↓

PATCH appends the third tape to the output. When this is completed PATCH types:

IS ADDITIONAL CODING DESIRED? NO↓

When the user has answered "N" for no, PATCH will punch a

checksum for the combined tape, output trailer, and then halt.

3. Patching

Patching is the means by which the user changes the binary data on the input tape. He does this by typing an address followed by the new contents of that address very much as with ODT. Before undertaking a general explanation let us consider a simple example.

Suppose the user has a binary program tape on which he wishes to change the contents of location Ø537 from 7ØØ4 to 71Ø4 and the contents of location 3166 from 77ØØ to 765Ø. He proceeds as shown below.

```
HOW MANY INPUT TAPES? 1⤶
TYPE ADDRESS OF FIRST PATCH: 537⤶
```

PATCH begins reading and punching core page by core page until it has read the page containing location Ø537 into its buffer. PATCH then enters "patching mode."

```
BUFFER CONTAINS Ø4ØØ-Ø572.
*537/7ØØ4 71Ø4⤶
*↑P
TYPE ADDRESS OF NEXT PATCH: 3166⤶
```

PATCH will now punch out locations Ø4ØØ-Ø572 with location Ø537 modified to 71Ø4. It continues reading and punching until

```
BUFFER CONTAINS 3ØØ2-3176.
*3166/77ØØ 765Ø⤶
*↑P
TYPE ADDRESS OF NEXT PATCH: ⤶
```

PATCH punches this new version of 3ØØ2-3176 and continues copying the rest of the input tape.

```
IS ADDITIONAL CODING DESIRED? NO⤶
```

The output tape is completed as usual.

There are many features of the patching operation not covered by the above example, but the example can serve as a basis for our discussion.

Specifying the First (Next) Patch Location:

When responding to the question, "TYPE ADDRESS OF FIRST (NEXT) PATCH:", the user does not have to type the exact location he wants. Any location on the same core page as the desired location will do. Furthermore, once this page is read into the buffer he may change as many locations on this page as he chooses and/or apply any of the special options to this page.

One important difficulty must be explained by an example. Suppose the input tape starts with an origin at Ø2ØØ, followed by 6Ø

data words, then a reorigin to Ø3ØØ followed by 6Ø more data words, including 7ØØ4 at location Ø342. The action will proceed as follows.

```
TYPE ADDRESS OF FIRST PATCH: 342↵
BUFFER CONTAINS Ø2ØØ-Ø257.
*
```

At this point the buffer actually represents locations Ø2ØØ-Ø377, though it only contains data up to Ø257. The data for Ø3ØØ-Ø357 has not yet been read into the buffer. The user should proceed as shown below.

```
*↑P
TYPE ADDRESS OF NEXT PATCH: 342↵
BUFFER CONTAINS Ø3ØØ-Ø357.
*342/7ØØ4 71Ø4↵
*↑P
```

If the user had placed the 71Ø4 in location Ø342 of the first buffer it would have been effectively nullified by succeeding data on the output tape.

The Buffer Contents:

The PATCH 2ØØ (octal)-word buffer always represents a single page of PDP-8 core. While patches are being made on a given page of data patches may not be made in locations off this core page.

Prior to reading data into any newly designated page the buffer is always initialized to contain all NOP's (7ØØØ). Furthermore the page delineators are set to a null page condition.

The page delineators are the first and last locations in actual use on a given core page. These are set when a page of data is read into the buffer and they may be modified during patching. On output PATCH punches out all data from the first to the last locations and no more. It should be noted that if the data on the page has been patched in a non-contiguous way, NOP's will be punched in the unused locations between the page delineators.

Consider an example. Suppose that the input tape contains an origin of 3ØØ2 followed by data through location 3162. After the read-in PATCH types

```
BUFFER CONTAINS 3ØØ2-3162.
*
```

The page delineators are thus set. If the user now types "↑P" only locations 3ØØ2-3162 will be punched. However, if the user patches location 317Ø, the page delineators are reset to 3ØØ2-317Ø and locations 3163-3167 all contain 7ØØØ. When the user now types "↑P" everything from 3ØØ2-317Ø is punched including the NOP's.

Patching Features:

(1) Slash not preceded by an address opens the last previously opened address. Such a slash at the beginning of patching a given page opens the location mentioned in response to "TYPE ADDRESS OF NEXT PATCH:"

(2) Negative numbers are acceptable as data.

(3) Line feed closes the current (or last opened) location and opens the numerically next location.

(4) ↑(shift-N) closes the current location and opens the numerically preceding location.

(5) Carriage return merely closes the current location.

(6) ↑P closes the current location and terminates patching on the current page. It automatically begins output.

(7) If a location is opened and then closed without being patched, its contents are not changed. Furthermore the page delineators are not changed unless actual patching takes place.

Example:

```
TYPE ADDRESS OF FIRST PATCH: 3105↓
BUFFER CONTAINS 3002-3162.
*/7402 7600↓
*↓
3106/1053↓
3107/3065 3072↑
3106/1053↓
*3002/0100 -2↓
*3177/7000↓
*3377/?
BUFFER CONTAINS 3002-3162.
*3176/7000 7000↓
3177/7000↓
?
BUFFER CONTAINS 3002-3176.
*3175/7000 7200↑P
```

PATCH will now punch out locations 3002-3176.

Patching Options:

Whenever PATCH types an "*" the user may request any one of the special options listed below merely by typing its first letter.

I Insert
D Delete
K Kill
M Move
F Field

These options are fully explained in section 6 of this document.

Errors:

Whenever the user makes an error (i.e., types an illegal character) during patching, any currently opened location is closed without modification.

4. Creating a Binary Tape

PATCH allows the user to create binary program tapes directly from octal data. This is done by merely specifying Ø input tapes. The technique is shown in an example below.

```
HOW MANY INPUT TAPES? Ø↓
*2ØØ/7ØØØ 76ØØ↓
Ø2Ø1/7ØØØ 1377↓
Ø2Ø2/7ØØØ 4776↓
Ø2Ø3/7ØØØ 1375↓
Ø2Ø4/7ØØØ 4776↓
Ø2Ø5/7ØØØ 52Ø5↓
*375/7ØØØ 313↓
Ø376/7ØØØ 4ØØ↓
Ø377/7ØØØ 317↑P
```

PATCH now punches leader code and the created data for locations 2ØØ-377.

```
IS ADDITIONAL CODING DESIRED? YES

*4ØØ/7ØØØ Ø↓
Ø4Ø1/7ØØØ 6Ø46↓
Ø4Ø2/7ØØØ 6Ø41↓
Ø4Ø3/7ØØØ 52Ø2↓
Ø4Ø4/7ØØØ 76ØØ↓
Ø4Ø5/7ØØØ 56ØØ↓
*↑P
```

PATCH now punches the data for 4ØØ-4Ø5.

```
IS ADDITIONAL CODING DESIRED? NO
```

At this point PATCH punches the checksum on the output tape, follows this with trailer, and then halts.

The method shown above is frequently used to create separate patch tapes for large programs. For this use it is very important that the user understand exactly how PATCH works. For each buffered page of code PATCH outputs all data in the buffer from the first address on the page where data is entered to the last such address. If there are addresses between the first and last addresses which have not had data entered in them, these locations will contain NOP's. These NOP's will be output along with the data that was actually entered. Thus in the above example, NOP's will be punched for locations 2Ø6-374.

The technique which should be used is shown below.

```
HOW MANY INPUT TAPES? Ø↓
*27Ø/7ØØØ 5777↑P
```

PATCH now punches leader code and 5777 at origin 27Ø.

```
IS ADDITIONAL CODING DESIRED?  YES
*377/7ØØØ 72Ø↑P
```

PATCH now punches Ø72Ø at origin 377. No NOP's are punched between 27Ø and 377.

```
IS ADDITIONAL CODING DESIRED?  YES
*72Ø/7ØØØ 1137
Ø721/7ØØØ 3616
Ø722/7ØØØ 71ØØ
Ø723/7ØØØ 5777↑P
IS YES
*777/7ØØØ 271↑P
IS ADDITIONAL NO
```

After PATCH punches Ø271 at origin 777, it punches the checksum and trailer and then halts.

It is useful to remember that there is no need to wait until the end of a PATCH query before responding.

5. Making Additions

After a user has copied one or more tapes onto the output tape, with or without patching, he is always given the option of making additions to the tape before the checksum is punched. This is done in the same way that separate binary tapes are created, except that the creation of the additions does not begin until all of the input tapes have been copied.

6. Special Options

Whenever PATCH is waiting for patching (signaled by having typed an asterisk) the user may request one of the special options below by merely typing its initial letter.

Field Expression (F)

If the user types "F", PATCH will type either "FIELD EXPRESSION READ FOR THIS PAGE WAS n. NEW FIELD =" or "FIELD EXPRESSION WAS NOT READ FOR THIS PAGE. NEW FIELD = ". In either case if the user responds with a single octal digit, Ø - 7, followed by a carriage return, PATCH will punch a field expression set to this new field at the beginning of the current page. Any past field setting will be omitted.

In the former case, where PATCH read a field expression from the input tape, the user may delete it by responding with a carriage return not preceded by a digit.

Move Page (M)

If the user types an "M", PATCH will type "MOVE THIS PAGE OF CODE TO THE PAGE WITH FIRST ADDRESS=", The user must then respond with an address on the page to which he wishes to move the current page of code, followed by a carriage return. When he has done this, the program will execute the move, type the new addresses of the page, and then wait for more patching.

Kill Page (K)

If the user types "K" PATCH asks "KILL THIS PAGE?" If the user responds to this with "Y" PATCH will kill the page in the current buffer and continue processing the input tape.

Delete (D)

If the user types "D" PATCH responds with "DELETE LOCATIONS". The user must respond as in one of the following two examples.

*DELETE LOCATIONS 200↵
*DELETE LOCATIONS 370-377↵

If the data deleted is at the beginning or the end of the page, the page boundaries are corrected. If the data is in the interior of the page, NOP's are inserted in place of the deleted data.

Insert (I)

If the user types "I", PATCH will type "INSERT THE FOLLOWING NUMBER OF WORDS:". To this the user must respond with a non-zero

octal number followed by a carriage return, PATCH will allow the insertion only if there is sufficient room at the end of the page to accomodate pushing down the code already in the buffer by the given number of words. If the given number of words will not fit, the message "INSUFFICIENT ROOM," is typed and nothing happens. If the insertion will fit, PATCH types "TYPE THE ADDRESS OF THE INSTRUCTION WHICH THE INSERTION IS TO FOLLOW:". The user must respond with an octal address followed by a carriage return. PATCH will then begin to request the words to be inserted by typing the correct addresses followed by slashes. The user must supply the octal word to be inserted at each address, followed by a carriage return. At the completion of the operation PATCH returns to wait for more patching of the program.

WARNING: The user is responsible for updating all references to addresses which have been pushed down.

NOTE: When this program was run on a PDP-8/I it worked except for address printout after carriage return. The following comment and patch to fix the bug were offered by Professor Howard Zimmerman, University of Wisconsin.

REGARDING THE BUG: The main bug must derive from the faster cycling time of the PDP-8/I compared to whatever computer this program was written for (maybe the PDP-5). After use of the slash option to get contents of an address and the line-feed option then to get the next contents, the next address has the first two digits overlayed on paper. A stall is already put into the program to allow time for carriage return but it is still not enough. The following patch corrects the problem. One could introduce a still longer stall without hurting.

PATCH:

```
LOC  1624/5370      JMP 1770
     1770/7001      IAC
     1771/7440      SZA
     1772/5370      JMP .-2
     1773/2156      ISZ PATAD
     1774/5225      JMP 1625
```