Technical Notes on DECsys

Bob Supnik, 19-Jun-2006

DECsys was DEC's first operating system for its 18b computer family and, indeed, DEC's first operating system for a computer smaller than its 36b timesharing systems. Introduced in 1965, DECsys provided an interactive, single user, program development environment for Fortran and assembly language programs.

System Requirements

DECsys requires an 8KW PDP-7 with dual DECtape drives. DECsys supposedly also runs on the PDP-4.

DECtape Usage

DECsys uses three different DECtape drives (units 1, 2, and 3), but only two are needed at any one time.

- Drive 1 holds the DECsys master tape as received from DEC. This is copied to drive 2 to create a working tape.
- Drive 2 holds the DECsys working tape. A working tape includes a full copy of DECsys as well as the user's programs and data. Each user has his or her own working tape.
- Drive 3 holds a scratch tape for use by Fortran and the assembler.

Thus, a dual drive system is workable. The drives are numbered 1 and 2 for creating working tapes, and then 2 and 3 during normal operation.

System/Working Tape Layout

A DECsys DECtape consists of 578_{10} (1102_8) blocks of 256 18b words each. The first seven blocks have a fixed format:

block 0	unused
block 1	tape label
block 2	system directory
block 3	library directory
blocks 4-6	keyboard monitor

Blocks 7-576 are available for system files, library files, and user files. Block 577 is not used. A minimal system tape, with Fortran, uses 143 blocks, leaving 434 blocks for user programs.

DECsys imposes multiple limitations on the size of users programs.

- A program can consist of no more than five separately compiled or assembled subunits.
- All subunits of a program must reside on the same working tape.
- A working tape can hold no more than 33 working files (minus directory space required for non-standard system programs).
- The upper boundary of a linked program must be below 11000_8.

Character Sets

DECsys uses a heterogeneous combination of FIODEC (the character code of the PDP-1), Baudot (the character code of the PDP-4), and ASCII (the character code of the PDP-7 itself). Regardless of the character set, alphanumeric data is packed three 6b characters per 18b word, as follows:

- FIODEC: all characters are naturally six bits
- Baudot: the 5b character code is shifted left 1, and the letter/figures flag appended as the sixth bit
- ASCII: the 8b character code is masked to 6b

The character sets used by DECsys are listed in the Appendix.

File Formats

Tape Label

The tape label consists of two Baudot strings. The strings are padded with 0's to an 18b boundary and terminated by a word of all 1's (777777_8). The first string is the tape label proper, the second the date. DECsys makes no use of the tape label other than to type it out if requested.

System/Working File Directory

The system/working file directory in block 2 has the following format:

word 0	directory length in words
words 1:254	directory entries
word 255	block number of first free block on tape

Directory entries are either system files or working files. System file entries are five words:

word 0	1 for system file
words 1:2	file name in Baudot, maximum 6 characters
word 3	starting block number of system program

word 4	starting address of system program

Working file entries are six words:

word 0	2 for working file
words 1:2	file name in Baudot, maximum 6 characters
word 3	starting block number of the Fortran version
word 4	starting block number of the assembler version
word 5	starting block number of the binary version

A block number of 0 indicates a file version that does not exist.

DECsys has no concept of data files. The Fortran compiler does not support access to file-based data sets.

Library Directory

The library directory in block 2 has the following format:

word 0	directory length in words
words 1:255	directory entries

Directory entries are variable length, depending on the number of entry points. If there are ***n*** entry points, the directory entry is 2***n***+3 words long:

words 0:1	entry name in Baudot, maximum 6 characters
words 2-3	second entry name (if any)
:	
words 2n-2:2n-1	nth entry name (if any)
word 2n	777777
word 2n+1	starting block number of the library file
word 2n+2	777777

File Storage

DECsys uses both contiguous files (for system files) and linked files (for working files). The working tape is assumed to be compacted, with all blocks before the first free block number (recorded in word 377_8 of block 2) in use, and all blocks beginning with the first free block number available.

Allocation of a contiguous file is straightforward. Contiguous files are always new; they cannot be overwritten in place. Further, DECsys requires the user to specify the length of the file (always a system file) in advance. Let ***f*** = the first free block number, and ***l*** = the length in blocks of the new contiguous file.

- The new system file will be stored in blocks [***f***,***f***+***l***-1].

- The new first free block will be ***f***+***l***.

Contiguous files cannot be deleted; DECsys has no file delete capability.

Allocation of a linked file is trickier. A linked file can be new, or it can be a replacement for an existing file. If a linked file exists, DECsys traces the existing links and overwrites the existing data. If the new file is longer than the previous version, new blocks are allocated starting at ***f***. If the new file is shorter than the previous version, any freed-up blocks are lost. DECsys has no capability to "compact" a DECtape, except by copying all the data to a different tape using UPDATE.

If the linked file is new, blocks are allocated starting at ***f***, just as with a contiguous file.

System File

System files are stored as contiguous files containing program text. The file starts with a two-word header:

word 0	2's complement of word count
word 1	initial load address – 1

The rest of the file is a contiguous stream of words to be loaded into memory. Note that the load address (stored in the file) and the starting address (stored in the directory entry) need not be the same, although they are often both 100_8.

Text File

Text files are stored as linked files containing FIODEC-encoded lines of text. Each DECtape block starts with a two-word header:

word 0	block number of next block in the file (0 if last)
word 1	2's complement of number of words used in this block (777402_8 if the block is full)

Text is then stored as 6b FIODEC characters. Four characters have special meanings:

14_8	end of line; next two characters are line number
15_8	end of page; next two characters are page number
16_8	end of file; next two characters are 0
17_8	master space (fill character)

End-of-line, end-of-page, and end-of-file must be aligned to a word boundary; the preceding word is filled with master spaces, if needed.

Library File

Library files are also stored as linked files, with the same two-word header as text files.

Keyboard Monitor Facilities

All calls to the keyboard monitor require that interrupts be disabled, and that extend mode be off.

Read Baudot String

This routine reads a string from the console Teletype and returns it, left-justified and Baudot-encoded, to the caller. The input buffer holds a maximum of six characters; excess input is discarded. Input terminates on almost any special character. The calling sequence is:

```
JMS 16261
Instruction to load buffer address      /XCT'd
..                                      /error return
..                                      /normal return
```

On a normal return, the two word buffer address contains the input string, and location 17103 contains the right-justified terminating character.

Of note is the use of XCT's, rather than LAC's, to obtain arguments. This was quite common in PDP-4 and PDP-7 software and makes the calling sequences difficult to read, since the inline "instructions" are in fact part of the subroutine and not part of the calling program.

Chain

This routine chains to a different system program. The calling sequence is:

```
IOF
LEM
16140:16141 = program name, in Baudot
JMP 16140
```

Appendix: DECsys Character Sets

Character	Baudot	FIODEC (S = shift)	6b ASCII
space	10	00	40
!	55	S 05	41
"	43	S 01	42
#	13	56	43
$	45	S 40	44
%	none	S 04	45
&	27	S 06	46
'	65	S 02	47
(	75	57	50
)	23	55	51
*	none	S 73	52
+	none	S 54	53
,	15	33	54
-	61	54	55
.	17	73	57
/	57	21	57
0	33	20	60
1	73	01	61
2	63	02	62
3	41	03	63
4	25	04	64
5	03	05	65
6	53	06	66
7	71	07	67
8	31	10	70
9	07	11	71
:	35	34	72
;	37	S 34	73
<	none	S 07	74
=	none	S 33	75
>	none	S 10	76
?	47	S 21	77
@	none	S 20	00
A	60	61	01
B	46	62	02
C	34	63	03
D	44	64	04
E	40	65	05
F	54	66	06
G	26	67	07
H	12	70	10
I	30	71	11

J	64	41	12
K	74	42	13
L	22	43	14
M	16	44	15
N	14	45	16
O	06	46	17
P	32	47	20
Q	72	50	21
R	24	51	22
S	50	22	23
T	02	23	24
U	70	24	25
V	36	25	26
W	62	26	27
X	56	27	30
Y	52	30	31
Z	42	31	32
[	none	S 57	33
\	none	40	34
]	none	S 55	35
^	none	S 11	36
_	none	S 03	37