MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo. No. 147

January 1968

A Multiple Procedure DDT

Thomas Knight

This Memo. describes a version of DDT used as the command level of the A.I. Group PDP-6 Time Sharing System (ITS). Special features include capability to handle multiple jobs, ability to stop on read or write references to a given location, and the ability of system programs to return command strings to be executed by the DDT.

General Features:

This DDT allows start up and debugging of several distinct programs at the same time. In any system of this sort, means must be provided to switch rapidly between programs and yet at the same time maintain the distinction between the states (running, stopped, etc.), the symbols, and the core images of the programs.

To achieve this distinction, several features are provided. The job$J command switches between jobs, preserving the state of the old job and restoring the state of the new one. If the programs being worked on are running at the same time they are being debugged, there is the possibility of the program running into trouble. In this case, the DDT does not wish to type a message out to you, since you are very likely doing work on this or another program which should not be interrupted. Under these circumstances, DDT types a bell, indicating that some job in some way wants attention. When the user wishes, he may then type $J at his convenience, and the DDT will respond with the program name, the reason for the break, etc. In this way it is possible to control more than one program at once without worrying about what effect one program will have on another.

Commands:

$A or $$A — sets absolute addressing mode (temporary or permanent).

sym$B — sets breakpoint at location "sym." Up to 7 breakpoints may be specified. A particular breakpoint may be set by sym$nB, where n is the desired breakpoint. When the program executes the breakpoint, the DDT types out "program counter>>$nB ."
0$nB removes breakpoint n.
$B removes all breakpoints.
sym(loc)$nB inserts a breakpoint at location sym and types out the contents of register loc when the breakpoint is encountered.

$C or $$C — sets constant mode (numbers in current radix)

$D or $$D — sets decimal type out mode

sym$E — searches current program for locations with effective address of "sym".
loc1<loc2>sym$E searches for such locations between loc1 and loc2.
sets . and the ring buffer of . to addresses found.

$F or $$F — sets type out mode to floating point.

$G or sym$G — starts the program at the current starting location or at location sym.
sym$$G — sets starting address to sym and starts at sym.
$@B is the pseudo-location where the starting address is stored.

$H or $$H — sets type out mode to half words(instead of instructions in the left half).

loc<n$I sets memory address register to location loc, with
n in the break register. Useful values of n are:
6 causes a program interrupt on read or write cycles of location loc
7 causes an interrupt on write cycles only. Note that some test instructions such as SKIPE do write cycles.
DDT types out MAR,addr>>instruction when the condition is met.
The instruction printed may be one after the instruction causing the interrupt.

name$J set current job to name.
create job name if no such job exists. Creation of new job types !

$J set current job to one that needs attention(in response to a bell that has been typed.). Types out the job name followed by the reason it needs attention.

name$$J set current job's name to name.

sym$K half kill symbol sym in the current job's symbol table.
(DDT will no longer type it out.)

sym$$K fully kill the symbol sym--will no longer
be accepted for input.

$$K kill all symbols of the current program.

$L n1 n2 user;dev:↵
or
dev$L n1 n2 user:↵
load file n1 n2 from device dev under user name user into the current job's core image. Zeroes the core, kills the symbol table, and closes IO channels before loading.
DDT remembers the name of the last file loaded or dumped, and uses it for unspecified arguments of the current command.
All BPT's inserted remain.

$$L as $L but does not zero core, kill symbols, or close IO channels.

n$M set mask for word searches to n

word$N search for a memory loc not equal to word (in the area not masked off by the word mask)
type out such locations, in the current mode.
stops typing out locations if any character is typed.

loc1<loc2>word$n searches between loc1 and loc2

$O or $$O set type out mode to octal

$P proceed from last breakpoint or interruption and assign tty to this job. n$P proceeds n times from this breakpoint before typing a BPT message.
$$P proceeds indefinitely., typing a BPT message each time the BPT is encountered
Typing any character during an indefinite proceed will stop it.

```
$Q      last quantity typed by either DDT or the user

$$R     set relocatable address mode.

n$R or n$$R     set current type out radix to n
                input radix is always 8. (or 10. for numbers terminated by .)

$S or $$S       set type out mode to symbolic (normal mode)

n$T or n$$T     set type out in bytes of size n

name$U  logs in user.  necessary before loading programs.
        Types out and deletes file SYS: name MAIL if it exists.

$U      loads new copy of DDT.  destroys all inferior procedures.

$$U     logs out user (kills jobs, deassigns all devices etc.)

$$V     lists jobs that DDT knows about, along with the state:
        P => proceedable (interrupt)
        R => running
        - => just loaded
        nB => broken on bpt n.

        * beside the current job, system index follows.

word$W  searches for locations equal to word in portions not masked by
        the search mask.  type out and effect on . as in $N.

ins$X   executes instruction ins in the current job's core image.
        not usable if the job is running.
        types 2 c.r.'s if ins does not skip, 3 if it does.

$Y      as $L but dumps instead of loads

$$Z     zeroes core of this job.
        loc1<loc2$$Z zeroes between loc1 and loc2.

|A

|B      turn on line printer output from DDT

|C      half kill the last symbol typed out and retype the last quanity

|D

|E      turn off line printer output

n|F     list files on tape n

dev|F   list files on device dev

n$$|F   flap tape n
```

|G full quit--to be used only when DDT hangs up completely.
not to be used to stop type out.

name|H load and start system programs--equivalent to

```
name$J
$L TS name SYS⏎
$G
```

types ! if a new job was created. Does not load symbols.

$$|H disowns current procedure.

|N

|O delete file--same format as $L

|P proceed from the last break or interrupt without
giving the teletype to the program. Allows the start up of several
programs or the examination and change of running programs.

|Q <=> ↑

|R

|S silences type out (empties type out buffer)

name|S set user name for inferior procedure

AIO|T dev: n1 n2 dev: n3 n4⏎
enter information into the device translation table.
a-atomic entry not further translated
o-output only
i-input only
translates opens for the first mentioned file to the second file.
* in the first entries indicate any file. In the second, specify
the same name as in the first.
trailing arguments are assumed *, lack of leading A,I,or O
implies IO.

|U untranslate--same format as |T.

|V turn on tty output

|W turn off tty output

|X stops current procedure from running. types PC and instruction
where it stopped.

$|X destroys current job. Requires a period typed following to
have effect.

$$|X destroys all jobs of this DDT. requires period.

n|Y flush current symbol table and load new one using n as an AOBJN pointer to user core.

|Z system wide quit character. when DDT is the superior procedure to a program, |Z acts as if a breakpoint were encountered at the current PC. Typing |Z on a console that is not in use loads a copy of DDT for use by the user.

@ =20000000, the indirect bit (IOR'S INTO WORD)

loc! open register, suppress type-out of contents.

" type last quanity in sixbit or ascii depending on whether $6" or $7" was last performed.

% symbol constituent

$(dollar sign) symbol constituent

' integer divide

& type last quanity as squoze symbol

(quantity) has value of quantity with right and left halves interchanged.

sym: defines symbol sym as equal to the currently open location.

value<sym; define symbol sym as equal to value.

sym$: sets program name to sym (used for relocatable programs)

* integer multiply

- mimus

+ full word plus

⌴ truncate following symbols to right 18 bits and add

= retype last quantity as current radix number

↵ close open register, deposit if register changed, close out temporary changes in type out mode.

$↵ opens last register in . ring buffer.

↓ close open register, and open next sequential one.

$↓ as $↵ , but register+1

↑ close open register, open previous one.

$↑ as $↓ but register-1

rubout erase last portion of a command or of new contents of a register.

. symbol constituent, or decimal pt. in decimal or floating pt. numbers. By itself, has the value of the currently open register.

, separate AC field from address in normal instruction type in.

,, separate left and right halves of word in half word type in.

/ type out contents of last address typed, and open the register.

? list undefined symbols

retype last quanity symbolically

[as / but types out constants

] as / but types out symbolically

$6]text(c.r.) treated as sixbit /text/

$7[!text! treated as ascii !text! ! => any character not in text

< or > separators of arguments to other commands

$6" or $7" set type out mode to sixbit and ascii characters respectively.

n$< or n$> takes non-zero fields of n and has value of $Q with those fields replaced by the new values.

\ open register addressed by $Q but do not change .

$\ has value of the contents of the address of $Q

⟶ open register addressed by $Q and change .

$⟶ open register addressed by left half of $Q and change .

typed after a previously undefined symbol, allows limited "automatic assembly"

Special Pseudo-Locations

.PC program counter for current job.

.MASK interrupt mask for current job.

.STOP stop bits of this job.

.PIRQC PI requests waiting for this job.

.IPIR as .PIRQC but IOR's word deposited into register.

.APIR as .PIRQC but ANDCAM's word deposited into register.

.PICLR 0 => PI in progress
-1 => PI cleared.

.MEMT top of user memory.

.MAR address and bits set in MAR

.MARPC last PC where MAR stopped program.

.IOC +n is contents of IO channel n

.IOS +n is contents of IO status word for channel n

.IOP +n is contents of IO pdl word n

Last Minute Changes (as of 1/5/68)

|H does not load symbols.

$|H as |H but load symbols. Without argument, loads symbols from last program |H'ed.

$; accepts arguments like $L and replaces symbol table of current job with the symbol table of the specified file.

$$; as $; but adds to current symbol table.

|Y now adds to symbol table, $|Y replaces table.

$$|O commands like $L. takes file name and executes DDT commands from that file.