

7/20/70

<b>bcc</b>	<b>title</b> PROCEDURES FOR TP PROGRAMS	<b>prefix/class-number.revision</b> PTPP/W-35.J	
<b>checked</b> <i>[Signature]</i>	<b>authors</b>		<b>approval date</b>   <b>revision date</b>   5/19/70
<b>checked</b>	R. R. Van Tuyl		<b>classification</b> Working Paper
<b>approved</b> <i>Mel</i>	<i>RRV</i>		<b>distribution</b>   <b>pages</b> Company Private   15

**ABSTRACT and CONTENTS**

Describes procedures for producing TP programs, i.e., assembling, loading, simulating, and preparing and punching selffill tapes.

Describes in detail how to produce a new DDTTP.

Quick description of SCOMB

My program Combination is a collection of small routines which were written as required to perform useful services. They are linked together by common subroutines only. There is no restriction on the type of subroutine which may appear in it now or in the future.

Currently the following routines exist.

	<u>Page</u>
Double-space	14
Prepare-paper-tape	9
Prepare-ITP-selffill	8
Finished	
Save-yourself	
ODDT	8
Continue-ODDT	
Symbols	7
Program	7
Clobber-file-copy	
DDT	8
Prepare-selffill-paper-tape	8
Place	
Verbose	
Quick	
?	2
Help	2
Initialize-swapper-tables	
Punch-paper-tape	11
Set-rubout-count	
Prepare-leader	12
Concatenate-files	13
Prepare-ITP-Loop	10
Single-Space	14
DIRECTORY	15

The dispatcher is the abbreviation lookup dispatcher as authored by Larry Barnes. Only the minimum subset of characters is needed to get into the routine. Furthermore, arguments may be supplied on the command lines or requested by the subroutines.

Rubout mode may be set to one of several modes:

Set-Rubout  $\emptyset$  - EXIT  
1 - GOTO DISPATCHER  
2 - Ignore first one  
N - Ignore first N-1 rebut

Rubout mode is initialized to 1.

Help or ? will print the dispatcher table.

Procedure for generating & loading TP Simulator with programs

The opcodes for TP are different. A file on (VAN TUYL) NRPTP will change NARP sufficiently to produce runnable code for TP. e.g:

@NARP.

```
Source ..... /NPITP,  
Obj ..... Nothing.  
@DUMP ..... '/ITP ASSM'.  
CONTINUE NARP (or /NARP.)  
Source..... /TEST.  
Obj ..... '/BIN'.
```

Will produce the appropriate binary.

The simulator source file is (VAN TUYL)SIMTP.

The simulator dump file is (VAN TUYL)DSIMT.

The dump file is prepared as follows:

@NARP.

```
Source ..... /SIMTP.  
Obj ...../IMTP.  
RESET.  
MACHINE SIZE IS 88K.
```

@DDT.

```
;T ...../IMTP.  
%F  
DUMP ON '/DSIMT'.
```

Now there is the problem of loading the simulator. It does not load binary files. Instead it loads a program space from a file and the symbol table space from a file. Preparing these files has been semi automated in a file called (VAN TUYL)SCOMB. To load a binary do the following:

@/SCOMB.

:ODDT  $\varnothing$  where  $\varnothing$  indicates carriage return

X;T /BIN. where X is the load point

etc

:

%F

:PROGRAM /PROGRAM  $\varnothing$

:SYMBOL /SYMBOL  $\varnothing$

:FINISHED  $\varnothing$

@

To load into the simulator and execute:

/DSIMT.

BEGIN;G

SET EXECUTIVITY PLEASE I>> .....

%F

@EXECUTIVITY -1.

@/DSIMT.

BEGIN;G

%LOAD /PROGRAM.

%USE TABLE /SYMBOL.

The Test processor 1 simulator document contains more information concerning the simulator.

Procedure for generating DDTTP

The following procedure will generate a program file of the current:

TP2ITP

@KDF.

&Read (VANTUYL)DDTTP;

&F.

\*\*\* Check ITPFG for -1

\*\*\* Check ITPFG for being +1

\*Read /DDTTP.

1,:SYMBOL:-1 Delete.

\*/

SYMBOL IDENT

\*WRITE ON '/2DDTTP'.

\*F.

@ /NARPTP.

@ /ITPASSM

Source /DDTTP,

Source /DDTTP,

OBJ '/BDTTP'.

OBJ '/BDDTTP'.

@/NARPTP.

@/ITPASSM

SOURCE /2DDTTP.

SOURCE /2DDTTP.

OBJ '/2BDDTTP'.

OBJ '/2BDDTTP'.

@/SCOMB.

@/SCOMB.

!ODDT Ø

!ODDT Ø

;T /BDTTP.

;T /BDDTTP.

;T /2BDDTTP.

;T /2BDDTTP.

;U

;U

%F

%F

bcc

P/c-n.r

PTPP/W-35.1

page

7

: Program /Program  $\phi$

: SYMBOL /SYMBOL  $\phi$



Preparing a Selffill TP program

SCOMB contains a program which will allow one to load a program into ODDT, then puts a loader followed by the processed program onto a file. The processing of the program includes squeezing blocks of 9 or more zeros out of the output and putting parity bits on each character. When the loader finishes control is transferred to the location you specify.

@SCOMB.

:ODDT  $\phi$

X;T /BINARY. (where X > 211B)

.

.

.

%F

:PREPARE-SELFFILL-PAPER-TAPE<sup>1</sup> STARTING-ADDRESS<sup>2</sup> '/PUNCH'  $\phi$

:F $\phi$

@

The file /PUNCH now contains an image of the paper tape.

---

<sup>1</sup> P-S is enough typing now

<sup>2</sup> digit {digit} [(B!D) [digit]]

Prepare an ITP Selffill file

An ITP program may be prepared for selffilling in a way similar to the preparation of a TP program. The program space of ODDT is used for getting the program. The loader ends at 266B, be forewarned. The loader zeros those areas between the blocks of your program. It does not touch areas outside your program except  $\emptyset$  to 266.

PREPARE-ITP-SELFFILL    STARTING-ADDRESS<sup>2</sup>    OUTPUT-FILE

---

<sup>2</sup> digit {digit} [(B!D) [digit]]

Prepare ITP Loop

This routine punches a non-blank character followed by the first 100B words of DDT's(ODDT's) address space. This is useful for preparing M-30 bootstrap loaders.

```
!Prepare-ITP-LOOP OUTPUT-FILE Ø
```

Punching a Tape

A tape may be punched by the following command

```
! PUNCH-PAPER-TAPE  FILE NAME  $\emptyset$ 
```

The program is Paul Heckel's Punch program, details of which are described in "Paper Tape Reading and Punching".

Preparing a Leader

A readable punched leader may be made by using the following command:

```
!PREPARE-LEADER INPUT-FILE OUTPUT-FILE  $\emptyset$ 
```

the input may be from the teletype in which case editing is possible with A<sup>C</sup>, Q<sup>C</sup>, W<sup>C</sup>, R<sup>C</sup>. Control D terminates the edit from the teletype. Multiple blanks are treated correctly and all other control characters (eg., carriage return, line feed) are ignored.

Concatenating Files

Several files may be concatenated by the following:

CONCATENATE-FILES OUT-FILE IN-FILE IN-FILE - - - - - Ø

Any number of files may be concatenated. End of file characters are observed and not output if they occur at the end of the file.

Double Space a File, Single Space

This routine double spaces and formats a file for printing purposes. The formatting algorithm is simple. Subroutines which are bracketed by two blank lines are grouped together onto a page. Subroutines which are longer than one page cause the next routine to be placed on a separate page.

```
!DOUBLE-SPACE INPUT-FILE OUTPUT-FILE ⌀
```

```
!SINGLE-SPACE INPUT-FILE OUTPUT-FILE ⌀
```

Directory

As an aid to initial debugging on M-1, this routine lists the symbols and values from DDT's symbol table. The symbols are sorted alphabetically on symbols and numerically by value.

@DDT.

:T /BTEST.

;C '/SYMBOLS'.

%F

@ /SCOMB

! DIRECTORY /SYMBOLS OUTPUT-FILE  $\phi$