

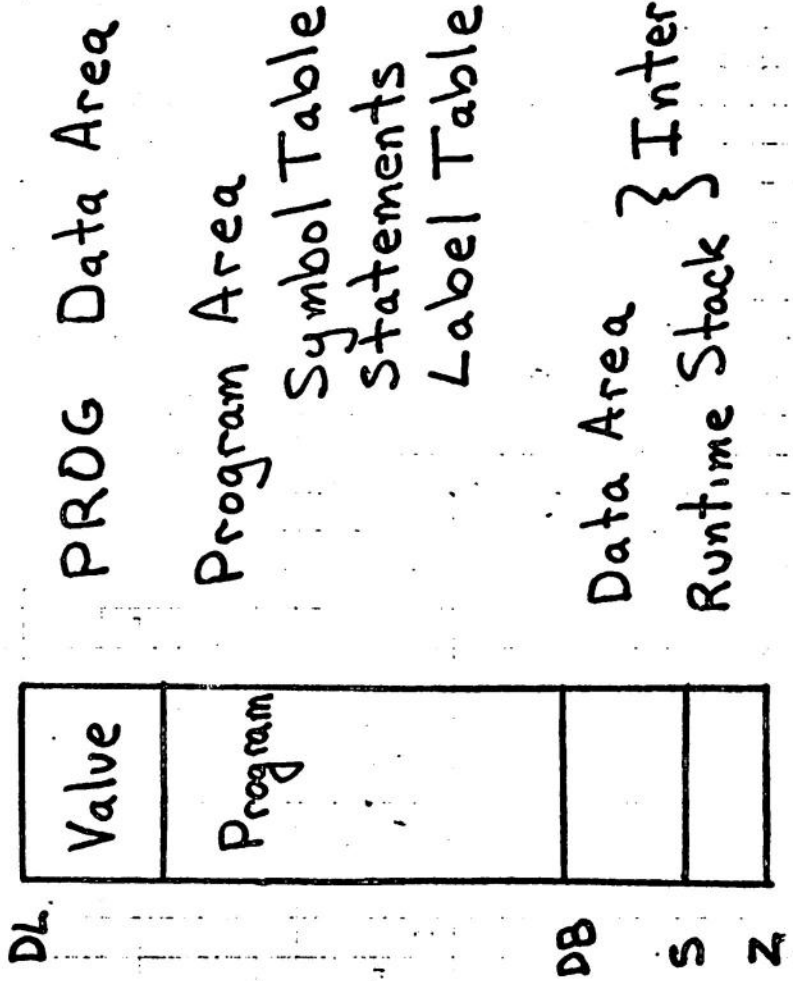
BASIC COMPILER - TERRY HAMM



# BASIC Interpreter Runtime Environment

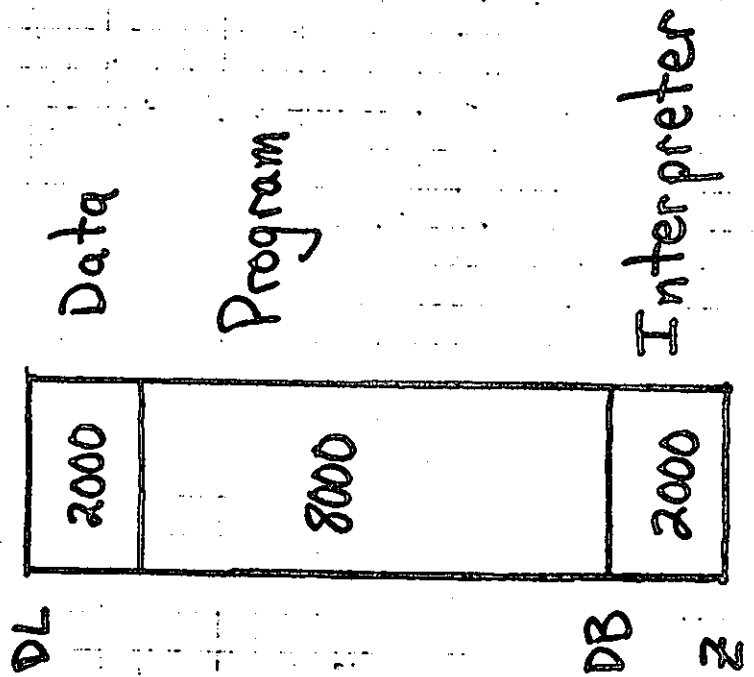
24 code segments ~ 37000 words

> RUN PROG



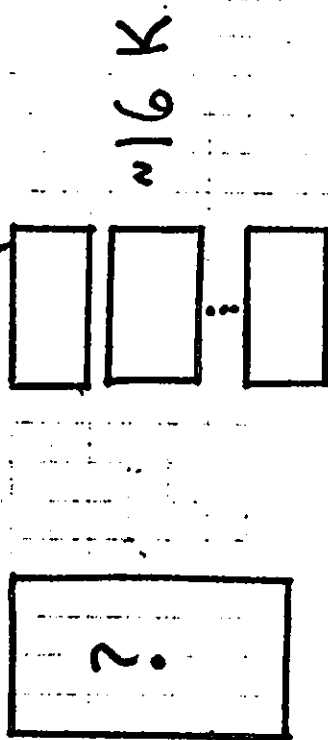
Data - 2000 words  
Arrays, variables  
file buffers

Program - 8000 words  
Interpreter - 2000 words

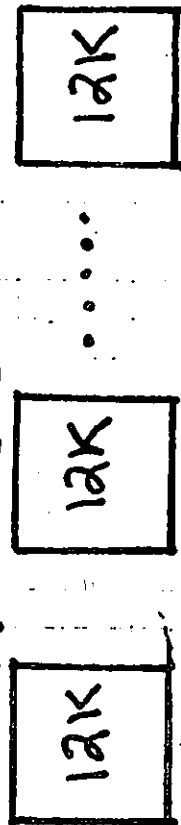


Total  
12K

MPE Interpreter Code



Data Stacks



$n$	Memory
2	40 K
4	64 K
8	112 K
12	160 K
16	208 K

$12 \cdot n + 16$

# FORTRAN

Program 9000 words  
3 segments

Data 2500 words

Code

3K

3K

3K

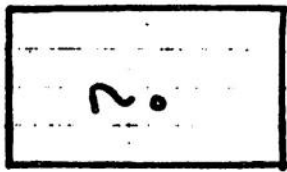
Stack

2.5K

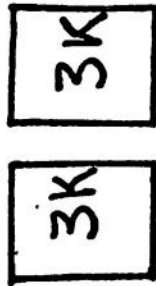
Total 11.5K

==

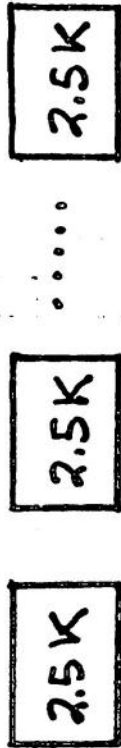
MPE



Program



Data Stacks



Memory

$\frac{n}{2}$

4

8

12

16

14 K

19 K

29 K

39 K

49 K

$$2.5 \cdot n + 9$$

$$\sim \frac{1}{4}$$





# HP 3000 BASIC COMPILER

# BASIC Compiler

- Extension of Interpreter
- Compatible to Interpreter
- Runs as subsystem under MPE
- Commands control the compiling

## Objectives

- Performance - improve BASIC program execution

1. Sharable Code
2. Smaller data stacks
3. Executable code

- Compatibility

- Generality

## User Interface

:BASICOMP [commandfile][, [uslfile][, listfile]]

default file

use

command file

\$STDIN

Input subsystem

uslfile

\$OLDPASS

commands  
output of

listfile

\$STDLIST

RBM's  
List output

:BASICPREP [commandfile][, [progfile][, listfile]]

:BASICGO [commandfile][, listfile]

## Subsystem Commands

\$CONTROL

parameter list

\$COMPILE

program name list

\$ENTRY

chain invoke list

\$TITLE

character string

\$EXIT

## \$CONTROL parameters

LIST

NOLIST

SOURCE

NOSOURCE

LABEL

NOLABEL

CODE

NOCODE

MAP

NOMAP

LINES = number of lines

WARN

NOWARN

USLINIT

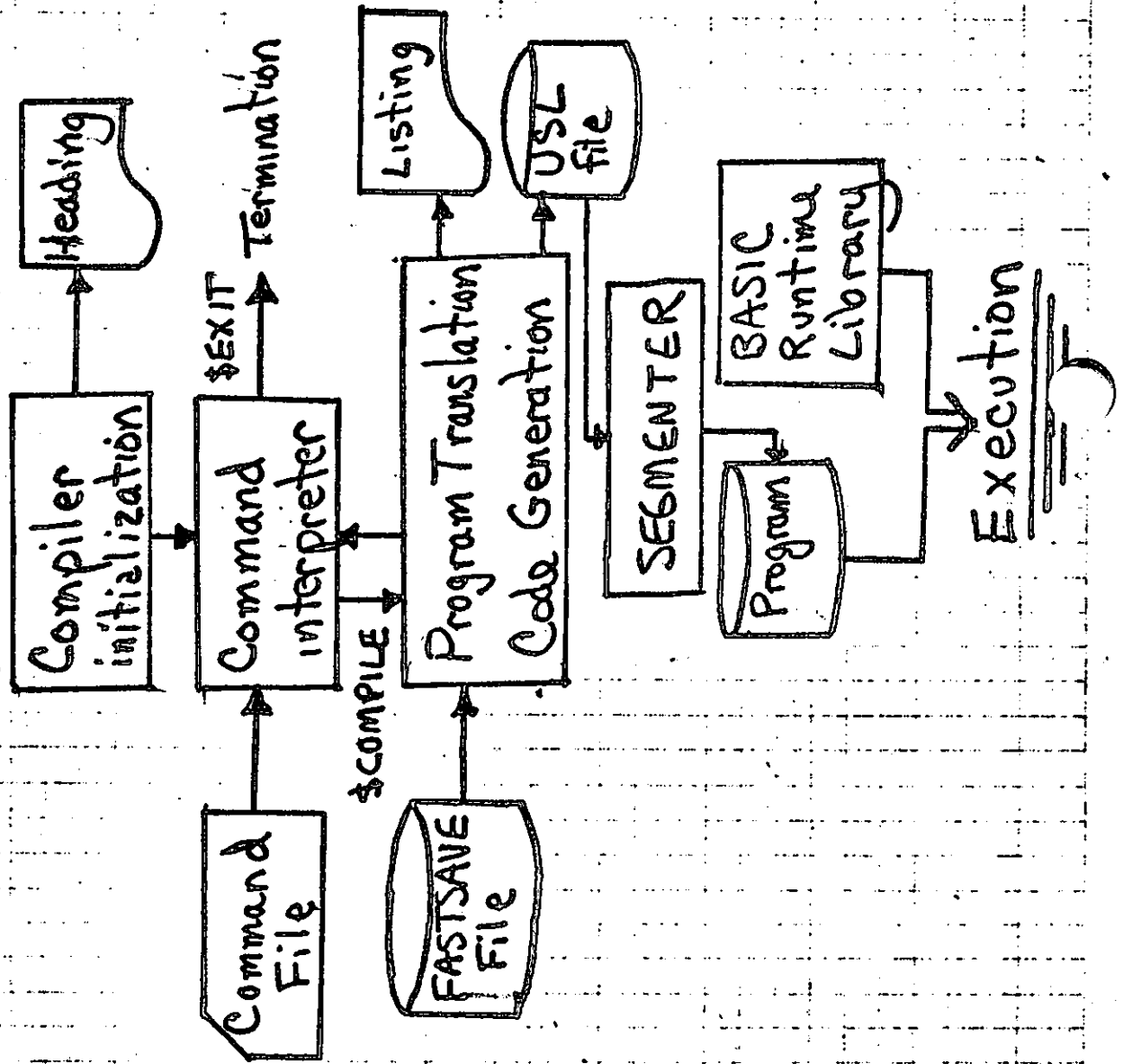
SUBPROGRAM START = name

SEGMENT = name

INIT

```
$CONTROL SOURCE,USLIMIT,NOWARN  
$CONTROL LABEL,LINES=50,START=C  
$TITLE "PROGRAM C", <<MAIN>> "JAN 25"  
$COMPILE C(100,200)  
$CONTROL NOSOURCE,NOLABEL  
$COMPILE A,B  
$ENTRY A,B  
$ENTRY SLPROG  
$EXIT
```

# Compilation Process





\$COMPILE A, B, C

SEG'

B'RUNOB
A
B
C

\$CONTROL SEGMENT = SEG1  
\$COMPILE A, B  
\$CONTROL SEGMENT = SEG2  
\$COMPILE C

SEG'

SEG1

B'RUNOB
A
B

SEG2

A
B

C
---

## B'RUNOB

- Open files BASICIN - BASICOUT
- Allocate file buffers
- Enable arithmetic & library traps
- Initialize runtime global variables
- INVOKE to START program
- Handle termination upon return from START

## BASIC runtime library

- COM area - file space management

- I/O utilities

Formatting

File manipulation

- String routines

- Matrix routines

- Builtin functions

# Runtime Stack

