

DARTMOUTH COLLEGE
COMPUTATION CENTER

PROCEDURE MANUAL

CCM- 5

Thomas E. Kurtz
15 March 1962

ALGOL - 30

SYSTEM OPERATING INSTRUCTIONS

INTRODUCTION

The ALGOL System for the LGP-30, called ALGOL-30, has three main parts:

1. Compiler
2. Loader
3. Interpreter.

Each of these parts is on a separate tape, and no more than one can be in memory at any one time. In addition there are several supplementary parts, each on a separate tape, as follows:

1. Reset (goes with the Compiler)
2. Hex Punch (goes with the Loader)
3. Dump (goes with the Interpreter)

These parts will be explained when the main part to which they are associated is explained.

The read program 10.4 is not used with ALGOL-30; in fact, no read program of any sort is ever permanently in memory. All tapes are bootstrapped in, and each of the three main parts has a bootstrap that is easily obtained. All tapes must be loaded through the photoreader.

A bootstrap may be obtained in any one of three ways. If one of the three major system tapes is in memory, a bootstrap may be easily obtained; see instruction in sections 1, 2, and 3. If 10.4 happens to be in memory, load the desired tape in the photoreader and perform the following

```
FLEX, OCNS, ;0000000, START, ,0000003,  
START, i0000, START, c000j, START,  
i0000, READER, START.
```

If the LGP-30 is essentially empty, then a bootstrap is obtained as follows:

```
FLEX, ONE OP, MANUAL, c0000, FILL, i0000,  
ONE OP, EXECUTE, MANUAL, c0004, FILL, c000j,  
ONE OP, EXECUTE, MANUAL, c0008, FILL, i0000,  
ONE OP, EXECUTE, CLEAR, NORMAL, READER, START.
```

Bootstrapping is a standard device, but it is emphasized that the bootstrap used here is a short photoreader version located at the low end of memory.

Once any of the system tapes is in memory all communication with it is through location 0000. The symbol OCNS of course

stands for the sequence "ONE OPERATION, CLEAR COUNTER, NORMAL, START."

A photoreader is essential to the use of ALGOL-30. A high speed punch is optional. The words FLEX and READER refer to the input toggle on the photoreader. (This report will assume flexowriter output; those installations using the high speed punch will make appropriate adjustments in these instructions, e.g., depressing BKPT 32.)

THE COMPILER

The Compiler, or Translator, takes programs punched on tape, in the ALGOL-30 language and produces an object tape in a semi-symbolic code called SNAP. Input to the compiler must be through the photoreader. Output may be via the flexowriter or the high speed punch; in the latter case, depress BKPT 32.

1.1 Loading the Compiler

Load the Compiler tape in the photoreader, obtain a bootstrap, turn to READER and press START. If a check sum error is encountered, back up the tape to the previous block of 20 spaces and press START. If a faster load is desired, depress TRANSFER CONTROL to ignore check summing. After loading the tape, be sure TRANSFER CONTROL is lifted since it is used for other purposes at other times.

1.2 Compiling a Program

Place the ALGOL-30 program to be compiled in the appropriate input device. Then we do the following:

- a. READER,
- b. PUNCH ON - TAPE FEED a foot or so,
- c. 6-BIT ON,
- d. Typewriter set to print about 7 inch lines,
- e. OCNS will then start the compiling.

1.3 Resetting the Compiler - Batch Compiling

After a program has been compiled, another one may be compiled after resetting the Compiler. Load the Reset Tape in the reader and perform the following:

- a. READER,
- b. TRANSFER CONTROL on,
- c. 6-BIT off,
- d. OCNS,
- e. TRANSFER CONTROL off,
- f. START.

1.4 Diagnostics

The Compiler contains checks for a number of common grammatical errors. If an error is generated, the source program tape must be recut and the compiling started anew. The nature of the compiler renders it impossible to correct errors in the middle of a compilation.

The error stops and possible causes are:

<u>ERROR CODE</u>	<u>CAUSE</u>
x var	too many variables
x dec	too many type declarations or too long a statement
x sym	too many symbols in a statement
x lab	too many statement labels (or too many neutral blocks)
x temp	temp storage exceeded- try shorter statement
x arr	too many arrays (15 max.)
def 2	label defined twice
./ type	wrong type on *
bool type	non-bool variables in boolean expression
:= type	mixed bool. and no. types on :=
sbscr type	bool. type for subscript of array
x dgt	too many digits in constant
x no.	too many constants (61 max)
ovflo	floating pt. max. no. size exceeded
def a	array not declared
undef xxxx	label xxxx used in goto but not used as statement label (illegal transfer into for statement)
sntax	illegal string
stop in 0028	operation not in package or done.

1.5 Obtaining a Bootstrap.

To obtain a bootstrap, simply lift 6-BIT, depress TRANSFER CONTROL, and perform OGNS. Lifting TRANSFER CONTROL and pressing START will cause the bootstrap to be executed.

2. THE LOADER

The Loader loads ALGOL-30 object tapes and certain function tapes in relocatable hex. The Loader converts the semi-symbolic language into absolute machine language, automatically links procedures compiled at different times, and performs several other functions.

2.1 Loading the Loader

Load the Loader tape into the photoreader and obtain a bootstrap. Turn to READER and press START. If a check sum error is encountered, back up the tape to a previous block of 20 spaces and press START. Depress TRANSFER CONTROL to ignore check summing for a faster read in.

2.2 Loading Programs

Loading programs, whether ALGOL-30 object tapes or relocatable hex tapes, is done as follows:

- a. FLEX,
- b. OCNS,
- c. Type "load",
- d. START (a bkpt 32 stop will be reached),
- e. READER,
- f. Load object tape in photoreader,
- g. START (loading now taking place),
- h. If there are additional tapes in your program, go back to step f, making sure the second pass of the loading phase is completed before pressing START again.

2.3 Load Completion Check

When the loading is completed, you may check to see if all procedures called for have actually been loaded as follows:

- a. FLEX,
- b. OCNS,
- c. Type "done",
- d. If typewriter prints "ok", all procedures called for have been entered, and control is passed to the breakpoint in step d of section 2.2.
- e. If some procedures are missing, they will be listed by name, and then control will pass to the breakpoint in step d of section 2.2. They may then be loaded according to section 2.2.

2.4 Hex Punching a Program

If you are sure your program is correct and has been correctly loaded, you may want to hex punch it for later use. If the Hex Punch routine is in memory (it can coexist with the Loader,) proceed to step h below; otherwise:

- a. Load Hex Punch tape in photoreader,
- b. FLEX,
- c. OCNS,
- d. type "bootstrap",
- e. START (a bkpt 32 is reached),
- f. READER,
- g. START (Hex Punch tape now reading in),
- h. FLEX,
- i. OCNS,
- j. Type "punch",
- k. START (a bkpt 32 is reached),
- l. Set up the punch,
- m. START (punching now taking place),
- n. When done, there is an unconditional stop before passing control to location 0000.

2.5 Batch Compiling

Batch compiling may be facilitated by batch loading - and - hex punching to reduce the number of times the interpreter and loader tapes must be handled. All that is required is to have the Loader and Hex Punch tapes in memory simultaneously, and then alternately load and punch ALGOL-30 object programs. The hex punched tapes are then individually bootstrapped in the memory at run time.

2.6 Obtaining a Bootstrap

To obtain a bootstrap for loading another part of the system:

1. FLEX,
2. OCNS,
3. Type "bootstrap",
4. START (stops at bkpt 32),
5. READER (load desired tape),
6. START (loading now taking place).

3. INTERPRETER

The interpreter performs all arithmetic, including integer arithmetic, handles input and output, keeps track of the return-jumps, and performs all the necessary bookkeeping operations. The only sequences that are done in machine language are the Boolean operations.

Communication with the interpreter is through location 0000, and the user can invoke a number of control and debugging operations.

3.1 Loading the Interpreter

After placing the interpreter tape in the photoreader, obtain a bootstrap, turn to READER, and START. If a check sum error is encountered, back up the tape to a previous block of 20 spaces and press START. For a faster read in, depress TRANSFER CONTROL to ignore the check summing.

3.2 Starting Your Program

To start the computation of your program,

- a. FLEX,
- b. OCNS,
- c. Type "start",
- d. START (stops at bkpt 32),
- e. Make ready any data tapes and activate the appropriate input device,
- f. START (your program is now running).

The "start" command clears the pseudo-accumulators, the compatible register (see below), and the return transfer table. Control is then passed to the first instruction of your first-loaded program (location 3300 in memory.)

3.3 Compatible Output

If output is desired for later input in large problems, a compatible output form with stop codes instead of spaces may be produced. One sets the "compatible" register as follows:

- a. FLEX,
- b. OCNS,
- c. Type "compatible",
- d. START (another input light is obtained),
- e. Type "yes",
- f. START (you are now back at location 0.)

If you want now to start your program, type "restart", then press START (bkpt 32) START. Typing "start" will set the compatible register to "no", that is, give ordinary output without stop codes.

The compatible register may be reset in either of two ways. First, one may enter his program by typing "start". Second, one may type "no" instead of "yes" in step e. above.

3.3 Continuing Your Program

If the program is stopped for any reason, it may be continued by

- a. FLEX,
- b. OCNS,
- c. Type "continue"
- d. START (stops at bkpt 32),
- e. START.

Control is passed to the beginning of the next interpreted instruction, the one following the one being executed when the program was stopped. This command does not affect the compatible register.

3.4 Hex Loading

To load a hex tape produced previously by the Hex Punch of the Loader, as in batch processing,

- a. Load desired hex tape in photoreader,
- b. FLEX,
- c. OCNS,
- d. Type "bootstrap",
- e. START (a bkpt 32 is reached),
- f. READER,
- g. START (program now reading in).

You may now proceed normally to "start" the program.

3.5 Tracing

If the program appears not to be working correctly, a trace may be obtained by depressing TRANSFER CONTROL. All statement labels and all variables being assigned values are printed in source program symbolism. The numerical value being assigned is then printed, in integer or floating point according to the type of the variable being assigned. (Only the last five characters of each label or variable name are printed.)

This trace mode is extremely easy to read and is very useful since correlating it with the original ALGOL-30 source program is simple.

3.6. Dump

A memory dump for variables may be obtained by

- a. FLEX,
- b. OCNS,
- c. Type "dump",
- d. START (stops at bkpt 32),
- e. 6-BIT,
- f. START (input light comes on),

g. Type in original source program symbolism the name of the variable whose value you desire and press START. The value is then printed in integer or floating point as the case may be. If TRANSFER CONTROL is down, only the first appearance of the variable in an assignment statement will be processed. Otherwise, identical values will be printed as many times as that variable appears in an assignment statement in your program. (Dump works by scanning your program for the symbol you input, then determines the memory location corresponding, and prints its contents.) There is then a carriage return and the input light comes on again.

- h. Repeat step g. as often as desired.

If the Dump routine is not in memory, after step d. above "error x trs" will be typed. The dump tape may be loaded as follows:

- a. Place Dump tape in photoreader,
- b. FLEX,
- c. OCNS,
- d. Type "bootstrap",
- e. START (stops at bkpt 32),
- f. READER,
- g. START (the Dump tape is now being read in into the top end of memory).

If your own program is so large that there isn't room for the Dump tape, then "error dump" will be typed. If you still want to continue loading the Dump tape, press START; perhaps the data storage at the top end of your program is not currently needed.

3.7 Obtaining a Bootstrap

To obtain a Bootstrap, do the following:

- a. Load desired tape in reader,
- b. FLEX,
- c. OCNS,
- d. Type "bootstrap",
- e. START (Stops at bkpt 32),
- f. READER,
- g. START.

3.8 Error Stops at Run Time

A number of errors at run time are detected. A five letter code indicating the type of error is printed. (To continue your program after an error stop, see section 3.3)

<u>ERROR STOP</u>	<u>DESCRIPTION</u>
x trs	Some routine not in memory.
xorder	Instruction not in package
xnest	Return table overflow or underflow (blocks, for statements, or procedures too deep.)
print	Integer to be printed exceeds 99999 in absolute value, Real number has exponent in binary ≥ 128 on print.
div 0	Division by zero has been attempted.
ovflo	Range overflow on assignment statement or temporary hold.
x int	Integer ≥ 32768 in absolute value on a float order, Binary exponent ≥ 15 on a round order, Absolute value of either operand on a divide ≥ 32768 .
input	Range error for a real number on input.
power	a^b , a and b both real, $a < 0$, 0 to the 0 power, any type, 0 to a negative power, any type.
x exp	Range error on exponentiation (or on a power instruction of type real.)
type	m^n , m and n both integer, $n < 0$, with no float order following immediately after the power instruction. Reprogram.
x log	Argument of the logarithm ≤ 0 .

4. GENERAL HINTS

The only time the 6-Bit is supposed to be depressed is during compiling or while using the Dump at run time.

TRANSFER CONTROL (TC) has three functions. During the loading of a tape, having TC down will ignore the check summing. In the Compiler, OCNS with TC down will produce a bootstrap. In the Interpreter, TC down will produce a trace of the running program.

All systems are loaded via a bootstrap, which is obtained in the Compiler as above, and in the Loader and Interpreter by performing OCNS and then typing "bootstrap".