

B. Chapman

FERRANTI ATLAS COMPUTER

THE MANCHESTER UNIVERSITY ATLAS OPERATING SYSTEM

London Computer Centre,
68/71 Newman Street,
London, W.1.

Offices, Works and
Research Laboratories:
West Gorton, Manchester 12.

May 1961

THE MANCHESTER UNIVERSITY ATLAS OPERATING SYSTEM

1. Introduction

This operating system has been devised by Manchester University and Ferranti Ltd., as a result of discussions between Professor T. Kilburn, Dr. R.B. Payne, Dr. F.H. Sumner and Dr. D.J. Howarth.

Jobs are initiated on the computer by input of information on slow peripheral equipments. A job may comprise one or more documents, defined in the following section. A job will not be executed until input of the relevant document(s) from slow peripherals is complete. The input information is collected in "wells" in main store, and is also recorded on a system input magnetic tape, the function of which is to act as an extension to the input wells in main store, and to preserve a record of all slow peripheral input. When input from slow peripherals is complete, a job is available for compilation and execution; priority is accorded to them in the order of input, but the actual order in which they are run is dependent also on the availability of magnetic tapes, slow peripherals and main store. Compilers will operate on a "Load and Go" basis, programs being executed immediately after compilation unless other action is specified by the programmer. Corrections can thus be made easily in source language, and the recording on magnetic tape of a source language program is shorter (by a factor of three or more in typical cases) than a record in binary form of the compiled program.

Results formed during execution of programs for output to slow peripherals are collected in other wells in main store and subsequently transmitted to the output equipments. The system output magnetic tape is used as a record of the output and as an extension of the output wells. A third magnetic tape, the system dump tape, is used as a temporary store for active programs and, where necessary, as an overflow of the system input and output tapes.

The layout of all system tapes is the same, sufficient information being recorded to locate any information given the initial address on tape. Facility is provided for the programmer to use information from these tapes instead of repeating input on slow peripherals; if a long paper tape is input, it may be used again by referring to its location on a system tape. The location on the system tapes of all slow peripheral input and output documents, dumps, etc. is printed with the programmer's results.

When large amounts of input or output are involved, the programmer may use private magnetic tapes to record the information. By suitable specification at execution time, this facility can reduce the load on the system tapes and will improve the overall efficiency of the system.

The following sections describe the way in which information is input to the computer and the way in which the programmer may use the various facilities of the operating system.

2. Titles and Headings

The title by which input information is known consists of one line of printing following a heading such as

```
COMPILER (x)
DATA
JOB
```

where "x" may be

```
INTERMEDIATE INPUT
MERCURY AUTOCODE
FORTRAN
ATLAS AUTOCODE
```

If the heading of the input information is

```
COMPILER MERCURY AUTOCODE
(The title on one line)
```

then the information is a source program in Mercury Autocode language. More generally, the information obeys the rules of Mercury Autocode, and may therefore include data as well as autocode statements.

If the heading of the input information is

```
DATA
(The title on one line)
```

then the information following is data which is read by a program during execution, which obeys no rules known to the system.

The heading "compiler" does not itself initiate the appropriate compiling, which is only commenced when a "job" heading is read. If the heading is

```
JOB
(The title on one line)
```

then the information following is the job description. In general this information is optional. It is terminated by an end of tape marker, in the case of a separate steering tape, or by "compiler" or "data", in which case the title is not repeated. The "job" heading will normally precede the source program tape as follows

```
JOB
(The title on one line)
COMPILER MERCURY AUTOCODE
```

then the source program itself.

Further optional information may be included in the job description such as:

1. data and program tapes (input)
2. output equipments used
3. magnetic tapes
4. store required, computing time, etc.

These sections of the job description are described below.

3. Input

A program reads in data by means of extracodes which are effectively "read next character (or string of characters) from data tape n" where "n" is a decimal integer ($n \leq 127$). The programmer's number of the data tape is specified in the "input" section of the job description. This section begins with the word

```
INPUT
```

and is followed by a list of the data tapes used in this job, each preceded by the programmer's number, n e.g.

```
INPUT
  1 (the title of data 1 on one line)
  2 (the title of data 2 on one line)
```

where there are two data tapes known by the programmer's numbers 1 and 2. These may have been read into the machine on the same input equipment as the "job" tape, either before or after it, or on other input equipments. The programmer's number, n = 0, is reserved for the program itself (and may be used in the program to read in data which follows the program as part of the same tape). A separate steering tape might be

```
JOB
  (the title on one line)
INPUT
  1 (the title of data 1)
  0 (the title of the program tape).
```

In this case, the name of the compiler to be used is written at the head of the program tape. When the "job" is on the beginning of a data tape, the "input" section of the job description must include

```
SELF = (n)
```

where "n" is the programmer's number by which this data is known within the program.

If the input section of the job description is omitted, it is taken as if

```
INPUT
SELF = 0
```

were included, and the program following is compiled and executed.

Since all input is automatically copied to the system input magnetic tape, a programmer may read his tape in again, direct from this input tape (e.g. to make a correction). To do this, in the "input" section of his job description he writes

```
TAPE (a)/(b)/(c)
      (n) (the title of his input)
```

where "a" is the system tape "number"

"b" is the number of the 512-word block of tape, and

"c" is the line within the tape block where his input starts. His title is, of course, written on the tape at this point, but the title is specified again as a check.

4. Output

A program outputs data by means of extracodes which are effectively "print next character (or string of characters) on output n" where "n" is a decimal integer ($n \leq 127$). The output equipments are specified in

the "output" section of the job description. This section begins with the word

OUTPUT

and is followed by a list of the output mechanisms used in this job, each preceded by the programmer's number, n e.g.

```
OUTPUT
1 (the type of equipment for output 1) (m) BLOCKS
2 (the type of equipment for output 2) (m) BLOCKS
```

The type of equipment may be

```
LINE PRINTER
TELETYPE
CARDS
N.E.P. TAPE
FIVE-HOLE TELETYPE
GRAPHICAL OUTPUT
ANY
```

where "teletype" means a 7-hole teletype paper tape punch, "cards" means a card punch, "N.E.P. tape" means static-reading magnetic tape, and "any" means output on a line printer, teletype punch, or on N.E.P. tape.

The operators can control which equipments are used most by disengaging the other output equipments. "m" defines the limit of the output, and if the output exceeds m blocks of internal characters, the program is stopped. If the number of blocks of output is not specified, it is taken as "1 block". Further, if there is only one output used, the output section may be omitted, and this is taken as if

```
OUTPUT
0 ANY 1 BLOCK
```

were included in the job description.

When printed, the output information itself is preceded by

```
OUTPUT (n)
(the title of the job)
```

and output of monitoring information is always on output 0.

5. Tapes

If a programmer uses magnetic tapes directly in his program (by use of tape extracodes as distinct from using tapes in connection with input or output) then he specifies each tape used by two lines in the job description

```
TAPE
(n) (the title on block 0)
```

where "n" is the programmer's number of the tape. When a new tape is required, the appropriate two lines of the job heading are

```
TAPE FREE
(n) (the title on block 0).
```

In this case, the title specified is written on block 0 by the supervisor.

If a file extends over several tapes, this is specified by a modified "tape" heading

```
TAPE / (m)
(n) (the title on block 0)
```

where "m" is the number of the continuation. The programmer's number n is the same for all m. The final tape of this file has the heading

```
TAPE / (m) END
```

If a program involves extensive input, then the job is preceded by copying this input to a magnetic tape. To initiate this copying process the input is headed

```
COPY TAPE FREE
(the title on block 0)
```

where the title specified is written on block 0. If a previously used tape is employed, the heading is

```
COPY TAPE (b)
(the title on block 0)
```

where "b" is the number of the tape block.

(He must always begin at the beginning of a tape block).

Information may be read from this tape subsequently by specification of the tape and title of the information in the "input" section of the job description.

If a program involves extensive output then the output can be written on a private magnetic tape. This is specified in the "output" section of the job description as follows:-

```
OUTPUT
(n) TAPE FREE / (the type of equipment) (m) BLOCKS
(the title on block 0)
```

where "n", "the type of equipment" and "m" are as for direct output, and where the title specified is written on block 0. If a previously used tape is employed, the specification is

```
(n) TAPE (b) / (the type of equipment) (m) BLOCKS
(the title on block 0)
```

where "b" is the number of the tape block.

This private tape is printed by a steering tape containing

```
PRINT TAPE
(the title on block 0)
```

if the whole tape is to be printed, or

```
PRINT TAPE (a) / (b) / (c)
(the title of his output)
```

if one section of tape only is to be printed, from tape a, block b, word c.

6. Miscellaneous Information

Further information may be given in the job description to indicate:

- (1) the amount of core and drum store used;
- (2) the time the program is expected to run;
- (3) the number of drums the program requires for programmed drum transfers.

All three apply to the execution stage of the program, i.e. excluding input from slow peripherals, compiling and output to slow peripherals, but including drum waiting time and tape waiting time. These are specified by

```

STORE          s
EXECUTION      p.q  HOURS
or EXECUTION   p.q  MINUTES
or EXECUTION   p.q  SECONDS
DRUMS          d

```

where "s" is the maximum number of core and drum 512-word blocks of store in use within the program during the execution stage, "p.q" is a fixed point decimal number such as

```
EXECUTION 7.25 SECONDS
```

where the program is expected to run for not more than $7\frac{1}{4}$ seconds, (if the estimate for store used and execution time is exceeded the program is stopped), and where "d" is the number of drums the program requires to reserve for drum transfers in which the sector number of the drum is specified.

If the total execution time is significantly different from the actual computing time, because there is considerable tape and drum waiting time, the actual computing time should also be specified, e.g.,

```
EXECUTION 5 MINUTES
COMPUTING 30 SECONDS
```

If information is not supplied in the job description, then

```
20 store blocks (10,240 words)
4 seconds computing time
```

and, of course, 0 drums are reserved. Estimates of the computing and execution times are taken as being equal unless both are specified explicitly.

7. End of Tape Markers

The end of a section of tape is indicated by

```
* * * (x)
```

where "x" is Z, A, B, C or T.

The marker

```
* * * Z indicates the genuine end of the tape/stack of cards
```

```
* * * A indicates abandon previous incomplete section if any
```

```
* * * B indicates that a binary tape follows
```

```
* * * C indicates the end of a section, and that there is another
      section following on the same tape
```

```
* * * T indicates a temporary stop within a section.
```

The number of characters, n, on a binary tape may be indicated by

```
(n) * * * B
```

where "n" is a decimal number.

On reading the marker " * * * Z" the peripheral equipment is disengaged by the computer. When the operator next engages this equipment, a new section (with the appropriate heading and title) is read. The marker " * * * C" indicates the end of a section of tape but the equipment is not disengaged and the next section is automatically read.

On reading the marker " * * * T" for a temporary stop, the equipment is disengaged as for " * * * Z". However, when the operator next engages this equipment, a continuation of the current section (without a new heading) is read. Finally, on reading the marker " * * * B", the computer reads the information following in binary without testing for further end of tape markers.

An alternative method of specifying the continuation of a section of data, without use of the marker " * * * T", is by means of a modified "data" heading

DATA / (n)

where "n" is the number of the continuation or part of the section of data, e.g. for a program with data on two distinct paper tapes, the data may be headed

DATA / 1
(the title of the data)

and

DATA / 2 END
(the same title)

and each tape ends with the marker " * * * Z".