Tech Info Library

Apple III: Printing from BASIC and Pascal

Revised: 8/15/86
Security: Everyone

Apple III: Printing from BASIC and Pascal

==

Many times it is desirable to have a program send selective, processed output to the printer. Both Business BASIC and Pascal have the commands to handle this chore; however, none of the language reference manuals provide an example for printing to a printer where the syntax of the commands is illustrated.

When directing output to the printer, it is important to remember that Apple III's Sophisticated Operating System (SOS) treats everything as a file. In this sense, any output, whether to a printer or diskette, is handled identically:

1. Programs pass data to SOS as files.

2. In SOS, the SOS File Manager passes the file to the appropriate device driver.

3. The device driver passes the output to the selected device.

The device driver does the job of transmitting the output in the fashion that the device requires it. Input data follows the same path, only in reverse. Understand these passes of files through the operating system. SOS and the device drivers can handle the details; you must handle the concept.

Here is an example of printing to both screen and printer from a Business BASIC program. Without the delay subroutine in line 100, everything would appear to happen at once, so we have it in the program only to improve the sense of sequence during execution.

```
10  OPEN#1,".printer": REM Declare the printer as a file.
20  PRINT "This is a test.": GOSUB 100
30  REM Output goes to the screen.
40  PRINT#1 "This is a test.": GOSUB 100
50  REM Output goes to File #1, the printer.
60  PRINT "This goes to the screen.": GOSUB 100
70  REM Without file specification
80  PRINT#1 "This goes to the printer."
90  CLOSE#1: END: REM Close the file ".printer" and end.
```

```
100  FOR delay = 1 to 2000: NEXT delay: RETURN
```

The program could also use ".silentype" for the printer device driver.

The seperate statements OUTPUT#x and PRINT can be used in succession to direct output to device x.

For instance, to simply list a program to the printer, the following commands can be entered at the keyboard in direct mode.

```
OPEN#1, .printer  (Notice, no quotes are needed in direct mode.
OUTPUT#1          (Routes all subsequent output to File #1)
LIST
CLOSE             (CLOSE in place of CLOSE#1 will close all
                   files instead of ".printer" -- with no other
                   files open it's just easier to type.)
```

There are other useful commands using this concept which are variations of other familiar BASIC commands besides PRINT. They are GET#1, INPUT#1, READ#1, and WRITE#1. The Apple III can have as many as ten files open for input and output at one time, so, where "#1" is used in the examples, it could be "#7", and so on.

The following is an example of how to print both to the screen and printer from a Pascal program. Again, the delay procedure is not necessary.

```
program PRINT_OUTPUT;
  var OutFile: text;

  procedure DELAY;
    var Count: integer;
    begin
      for Count := 1 to 2000 do
    end; {Delay}

begin
  rewrite (OutFile, '.printer'); {Declare the printer}
                                 {as a file.}
  writeln ('This is a test.'); {Output goes to the screen.}
  DELAY;
  writeln (OutFile, 'This is a test.'); {Output goes to}
                                        {the printer.}
  DELAY;
  writeln ('This goes to the screen.');
  DELAY;
  writeln (OutFile, 'This goes to the printer.');
  close (Outfile) {Close the file ".printer" and end.}
end. {Print_Output}
```

The Apple III can have as many as ten files open for input and output at one time; "OutFile" is only one. Additional files can be declared with additional REWRITE statements.

To simply list a program to the printer, go to the Filer and transfer the text file to .PRINTER (for the Apple III) or to PRINTER: (for the Apple II and Apple IIe).

Since Pascal treats output as a file, both of these examples work with both Apple II and Apple III Pascal.

Apple Technical Communications

Tech Info Library Article Number:160