CHAPTER 7: CONFIGURING PRINTERS

Introduction

The most common method of transmitting data to a printer is by outputing to the LPT1 device driver. This is an internal device driver which means that, although it is treated the same as drivers which are loaded from a "DEVICE=" statement in the CONFIG.SYS file, no such statement is required to install it. Note that PRN is an alternate name for LPT1, and that LPT2 and LPT3 refer the second and third logical printer devices.

All of the logical printer devices (LPT1, PRN, LPT2, LPT3) exist even when no actual (physical) printer is connected. Indeed, when the spooling subsystem is used, you can "print" from applications even when your printer is down. Printer output will end up in disk files which may be reviewed online or saved for printing at a later date. Through its virtual machine emulation features, MOS can isolate logical printing from physical printing which allows a number of useful manipulations in the re-routing of printer data.

You may be aware of some applications which call the interrupt 17h services of the BIOS to perform printing. This is often done by applications which require a greater degree of control over the process since status conditions can be detected more precisely (e.g. printer offline or out of paper).

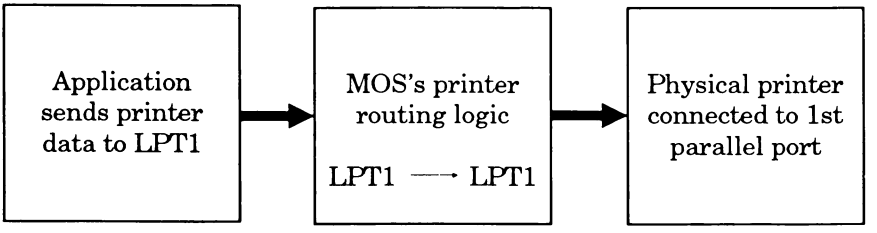
Whereas when running under DOS, calls to interrupt 17h bypass the operating system and interact directly with the BIOS, under MOS the response to an INT17 call is made by the operating system itself. Thus redirection control is maintained regardless of whether applications print through an LPT device or INT17.

CHAPTER 7

Routing

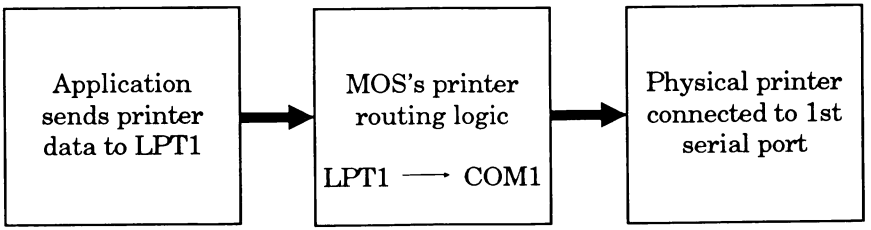
PC-MOS supports a variety of printer routing configurations. Data which is sent to the standard LPT1 device may be redirected to a COM port to support a serial printer. You can re-route printer data from one LPT device to another, from an LPT device to the printer port of a serial terminal workstation, or you can spool printer data into disk files which will be sent to the actual printer by another task. In addition, this logical-to-physical printer translation is task specific. Task 0 can be routing its LPT1 data directly to the first parallel printer, task 1 can be routing LPT1 to COM3 for a serial printer and task 2 can make the printer connected to its workstation printer port accessible through the logical LPT1 device.

The ROUTE subcommand of the MOS.COM system utility is used to define the route that printer data will take in PC-MOS. Each of the possible combination groups is shown along with the form of MOS ROUTE command required, implementation notes, and typical usage examples.



MOS ROUTE LPT1 TO LPT1

This is the normal configuration to which a task defaults when no other re-routing directive has been issued. The only time you would actually need to issue this form of the ROUTE command is when canceling a previous re-direction. Note that MOS ROUTE LPT2 TO LPT2 would be used to re-establish normal routing for the 2nd printer, and MOS ROUTE LPT3 TO LPT3 for the 3rd.



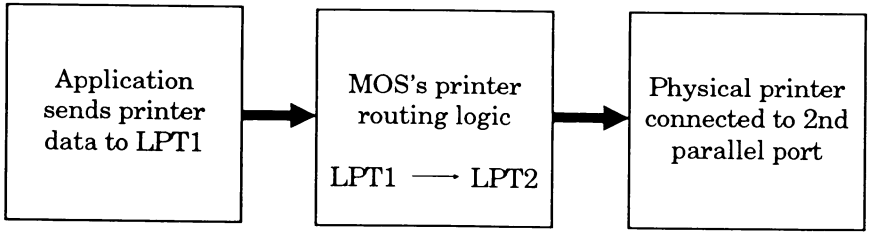
MOS ROUTE LPTx TO COMy
(where x can be from 1 - 3 and y from 1 - 24)

This routing could be used to give certain tasks their own printer and make each one appear as LPT1 to all applications run within the task. For example, task 1 could have a serial printer connected to COM5 and have a MOS ROUTE LPT1 TO COM5 command in its startup batch file. In a similar way, task 2 could have a printer on COM6 and use a MOS ROUTE LPT1 TO COM6 command. Note that any re-routing of LPT1 also affects all output sent to the PRN device.

Serial printers can be especially useful in a multiuser environment since their cables can generally be run farther than parallel ones. In addition, modems and fiber optic repeaters may be used to even further extend their range.

A software type of handshaking (XON/XOFF or XPC) will typically be required in such installations. For more details on handshaking and serial port initialization, refer to Chapter 8.

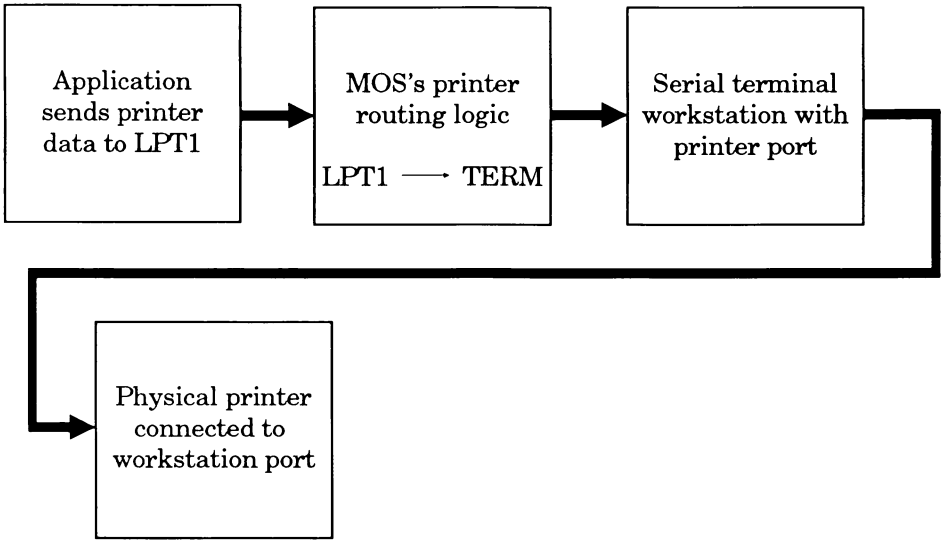
Some applications allow you to specify a serial or parallel printer through a setup menu or configuration file. Even though you could direct such applications to send their printer output directly to the COM port, its generally better to make them use LPT1. There is no speed penalty with MOS's printer redirection, and maintaining a consistent interface standard will reduce system maintenance chores. If you ever need to use a different serial port for the printer, change the ROUTE command in the startup batch file and all applications will be converted automatically.



MOS ROUTE LPTx TO LPTy
(where x and y can range from 1 - 3 and x < > y)

This directive could be used in a system with two users and two parallel printers. The first task would use the default routing (LPT1 -- LPT1) and have a printer connected to the first parallel port. The second user would have their printer connected to the second port and have a MOS ROUTE LPT1 TO LPT2 command in their startup batch file.

Applications run in the second task, which send printer output to LPT1, will end up printing on what is actually LPT2. This way, there is no need to specially configure a copy of each application to use LPT2. Indeed, few applications provide that level of customization. You could also use this routing if you have applications which are not re-configurable and your first parallel port is dedicated to some use other than LPT1 (e.g. LANLink 5X)



MOS ROUTE LPT1 TO TERM

This directive only applies to tasks associated with serial terminal types of workstations. The type of printer used and the cabling and handshaking requirements are between the terminal and the printer. Many terminals can be ordered with either a serial or parallel port as an option.

One important factor to keep in mind is that while printer data is coming down the host to terminal communications link, the workstation will be put into a dormant mode. To cancel a routing to a remote terminal's printer port use the MOS ROUTE NOTERM command. A MOS ROUTE LPT1 TO LPT1 (or similar) will also do this automatically.

Automatic Printer Arbitration

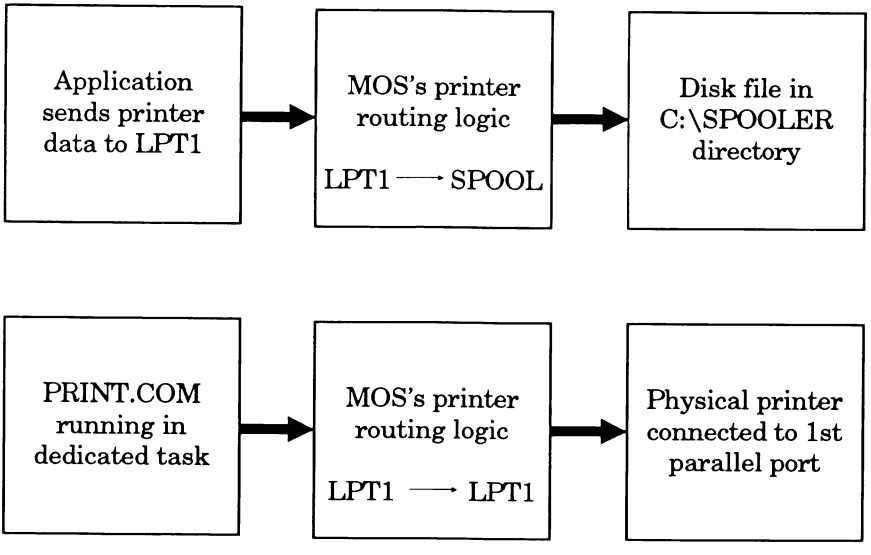
Beginning with release 4.00, MOS provides a new printer sharing alternative besides the spooling subsystem. This new method involves an automatic form of printer arbitration whereby the first user to access a printer retains control of that device until they cease to use it for a certain period of time. Any other users who try to access the device while it is under the control of the first user will receive an error status indicating that the printer is busy.

The main benefit of this approach is that it is much simpler to set up and use than the spooling subsystem which is described in the next section. It is made very easy since this automatic arbritration feature is applied by default. The MOS HOLD command can be used to change the reservation lock-in time from its default of 15 seconds. Aside from that, there is nothing that the user or the system administrator needs to do to begin using this feature.

One consequnce of this simplified approach is that submitting other print jobs when the printer is busy will not produce a mixed up listing, but instead result in a time out error for all but the first user. This would be true for the user who instigated the current print job as well as for any other users. The spooling subsystem does not have this limitation. Multiple users can submit multiple print jobs, at the same time, with no interruption in their work flow.

The Spooling Subsystem

The PC-MOS User Guide does a thorough job of explaining the general operation of the spooling subsystem, spool file naming and the meaning of the parameters for spool and print. This section will present a brief introduction to the spooling subsystem and then elaborate on some of the different installation configurations which may be used.



Print spooling is a type of routing where data originally sent to an LPT device driver is written to a temporary disk file (a spool file) instead of being sent directly to a physical printer. The contents of spool files are eventually sent to a physical printer by a system utility program which runs in a background task dedicated to that function.

It is possible to mix spooling with the other routing methods described above. For example, tasks 0 and 1 could each have their own private printer while tasks 3, 4 and 5 could all be spooling their output to one shared printer. It is also possible for one spooling subsystem to operate more than one printer and for more than one spooling subsystem to be active within a system. As with the ROUTE command, re-directing printer data to a spool file is task specific.

Two system utility programs, SPOOL.COM and PRINT.COM, work together to support the print spooling subsystem's operation.

The SPOOL program is run in each task which is to take advantage of print spooling. This system utility is a TSR (Terminate and Stay Resident) type of program which modifies the actions of the kernel with respect to printer I/O. SPOOL causes the kernel to translate an output to a logical printer device (e.g. the LPT1 device driver) into output to a temporary disk file (a spool file). This TSR also provides a pop-up menu which can be used to dynamically configure certain aspects of the spooling process.

By buffering printer data in spool files, MOS can easily control which task's output is sent to a printer. Even though multiple tasks may be generating printer output at the same time, all destined for the same physical printer, the isolation and access control provided by the spooling subsystem insures that no printer data is lost or interleaved.

The spooling subsystem component responsible for reading spool files and sending their contents to a physical printer is the PRINT program. The spool directory is periodically examined by PRINT to check for files which are ready for printing. To prevent unnecessary system loading, the PRINT program enters a state of suspension when no ready files are found. When in this state, PRINT will wake up periodically to do another scan of the spool directory.

Spool files are categorized by the three attributes: Disposition, Priority and Class. The first two characters of a spool file's extension delineate the Disposition and Priority respectively. A file's Disposition controls what happens to it once it has been printed while the Priority is usd to control printing order. The third character of the extension designates the spool file's Class. The use of this attribute is what our discussion here will focus on.

PRINT uses the spool file's Class to select which files to operate on, to route files to particular printers and to designate that hand fed forms are to be used. When two or more printers must be operated through MOS's spooling, there are three basic options:

1. Set up a single PRINT partition (one task running the PRINT program) and use Class designations to direct different files to their respective printer. When this method is used, only one printer will receive data at a time.

2. Set up one PRINT partition for each printer but configure each one to watch one common directory for spool files. Different Class letters must be used for each to prevent two PRINT programs from trying to grab the same file.

3. Set up one PRINT partition for each printer and configure each one to watch a unique directory for spool files. The Class letter restrictions of case #2 do not apply.

When a task generates a spool file, the Class attribute assigned to that file comes from the current Class setting in the SPOOL program. The Class to use can be declared to SPOOL when it is initially loaded and can also be changed at any time by popping up the spooler's menu window. In cases #1 and #2 the Class attribute of a spool file is used to direct the file to a particular printer.

Ln case #3, the directory specified when SPOOL is installed will determine the destination printer. Once a task's SPOOL program has been installed to send printer output files to a certain directory, this becomes a limiting factor since unlike the Class, the spool file directory cannot be dynamically changed. This is a fixed type of assignment which must be considered when designing a system.

There can be cases where multiple printers are being operated by a spooling subsystem and a fast response is required from one particular printer. Printing of customer receipts in a point-of-sale installation would be a good example of such a situation. By dedicating a print partition to the receipt printer, and possibly raising that task's priority, a good response time can be assured. Using a printer with a large RAM buffer will also have a significant effect on overall system throughput. A second print partition can be installed for users who will share one or more other printers to produce inventory and accounting reports and other documents.

Figure 7-1, on the following page, shows an example multiple printer configuration.

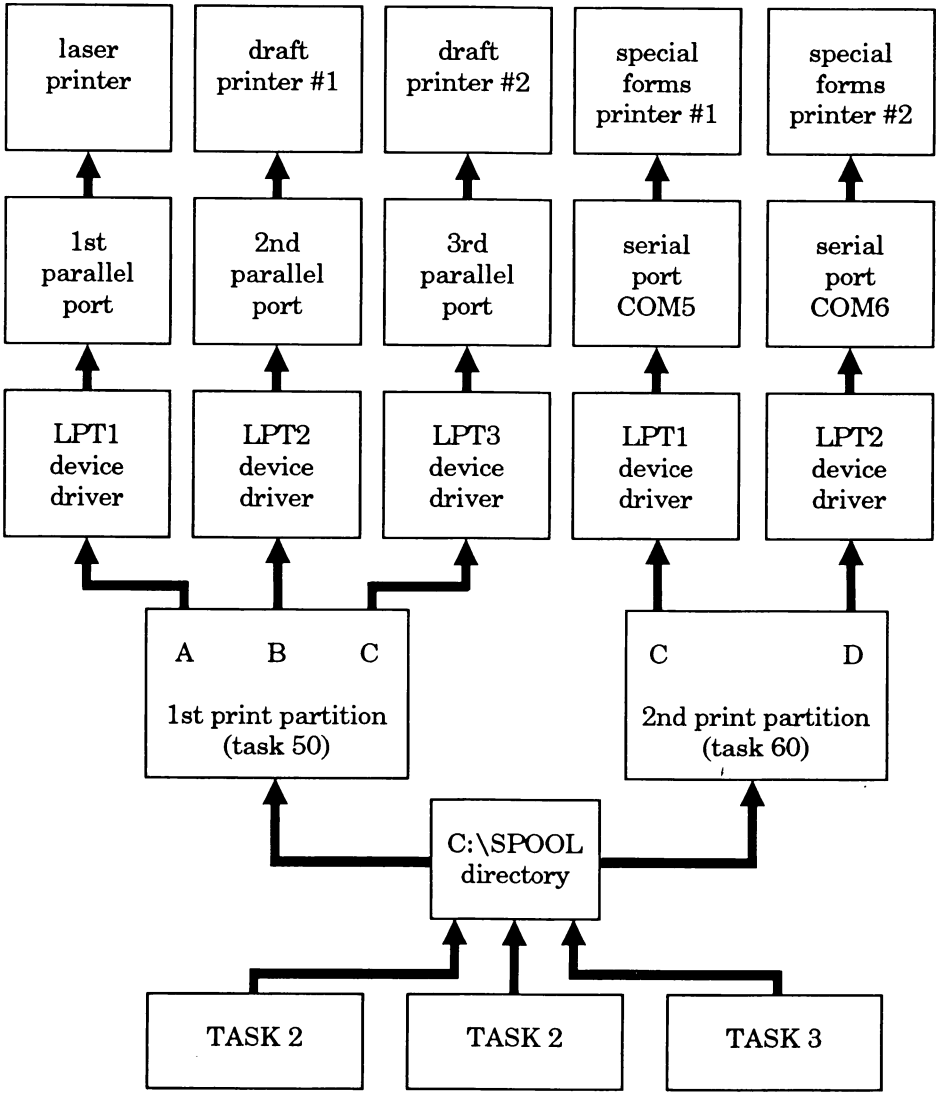

laser printer
draft printer #1
draft printer #2
special forms printer #1
special forms printer #2
1st parallel port
2nd parallel port
3rd parallel port
serial port COM5
serial port COM6
LPT1 device driver
LPT2 device driver
LPT3 device driver
LPT1 device driver
LPT2 device driver
A B C
1st print partition (task 50)
C D
2nd print partition (task 60)
C:\SPOOL directory
TASK 2
TASK 2
TASK 3

Figure 7 - 1

Figure 7-1 shows an example of a system configuration which might be used in a office where a moderate to large volume of printing is done. To reduce the amount of unloading and reloading of special forms, two small dot matrix printers are dedicated for forms printing. In the 2nd PRINT partition (task #60), the startup batch file would need to include a MOS SERINIT command to initialize the serial ports as well as following commands to establish the necessary printer routing:

```
MOS ROUTE LPT1 TO COM5
MOS ROUTE LPT2 TO COM6
```

Note how the logical-to-physical translation of an LPT device to an actual printer is specific to each task.

A user can direct their task's print output to any of the five printers by setting the proper class letter with the SPOOL program's pop-up menu. You must be careful, however, when changing the printer class during a series of print jobs. If you've just sent a print job to the Class D special forms printer and are ready to send your next job to the laser printer (Class A), you must be sure that the filename of the spool file for your Class D print job has been formed already. If the Class D print job is relatively small (less than 2k bytes) it may fit entirely within the spool buffer. In this case, the spool file won't always be created right away. If you change the Class setting with the pop up menu before the Class D job's file is created, this job will end up going to the laser printer.

The File Capture Technique

When you'd like to preview the layout of a new database report you are designing or wish to check the spacing of a word processing document, there's nothing to say that you have to let a spool file be picked up by the PRINT program and sent to a printer. By changing SPOOL's disposition to H (for hold) you can make PRINT ignore the file. Once the file has closed you can use RENAME to move the file out of the spool directory (REN/M must be used to move a file, see Chapter 5). You can now load this file into an editor and browse through it.

Another use of this trick is to capture screen display output in a text file. By keying Shft-PrtSc with SPOOL installed you can end up with a file containing your current display screen. In addition, keying Ctrl-P from the command prompt can be used to direct a copy of all subsequent display output to the current logical LPT device (a spool file in other words). Note that this will not work for text which is displayed through direct writes to video memory --it is most useful for recording a command line session.

If you are not presently using the spooling subsystem but wish to take advantage of this technique, install SPOOL in your task and specify a miscellaneous directory for its output. When you are done capturing screen output or other print output in files and wish to establish your previous direct access to your printer, pop up SPOOL's menu and change the disposition to N to deactivate spooling.