Printing to a network printer

Thanks to Johnny Billquist for the help to get this to work.

In the late 1970s printing was very different from what it is today. ASCII text files were sent to the printer and were printed there using whatever font the printer was built to use (sometimes even mechanical such as in a ball type typewriter). BSD Unix systems such as 2.11BSD used the line printer daemon protocol lpd. With the arrival of the first DOS systems a new problem arised: Most inexpensive printers supported the DOS newline convention (\r\n), whereas Unix used (\n) and early Apple systems used (\r). But it was often possible to set the ancient printers up to use any one of these. Fortunately, many modern printers still support the ancient lpd protocol using port 515, but I have not been able to turn on Unix style newline handling on my networked HP OfficeJet Pro.

The BSD Unix printcap configuration file allows to define a filter, which converts for example the newline characters in a text file, using the if or of entry. But unfortunately it appears that if and of is only used for directly connected and not for remote printers. The reasoning behind this might have been that the conversion was supposed to be done on the remote and not the local system. Remember that remote printer in those days meant a printer hooked up to a remote Unix system, and not a printer with a built in printer server as today. In my setup I have therefore to use a workaround to implement the conversion. Please let me know if you are aware of a better solution.

Let us now install a network printer in 2.11BSD

Login as root, and add the ip address and name of your printer to /etc/hosts, for example:

```
192.168.1.11 officejet.home.lan officejet
```

Instead of officejet, use whatever name you want to give to your printer.

In /etc/printcap, put a # in front of all lines of existing printers, and add for example at the bottom:

```
lp|officejet:\
      :lp=:sh:rp=text:sd=/usr/spool/lpd:\
      :rm=officejet:\
      :lf=/usr/adm/lpd-errs:mx#0:
```

lp is the standard printer for lpr. This defines therefore that the networked officejet is the standard printer.

Now reboot so that the system reads the /etc/hosts file.

Log in as a user, find a short text file and print it with

```
lpr your_text_file.txt
```

If everything works fine, you are all set. Enjoy printing on 2.11BSD.

If the printer prints line feeds, but no carriage returns, as for example HP printers will do, we need to install a filter to add the carriage returns. The printed output in this case looks like

```
   line number 1
                  line number 2
                                      line number 3
```

Installing a filter to insert the necessary \r characters

Login as the user who wants to use the printer. This could of cause also be done using root and installing the executables in a shared directory, but I want to keep the basic 2.11BSD as much as possible in its original state.

Transfer the todos directory from this repository to your 2.11BSD home directory. Compile todos.c there with “make”.

Copy the shell script print and the todos from the todos directory in this repository to your ~/bin directory. The source todos.c and the Makefile is also available for inspection and modification. Log out and back in.

The shell script print pipes the file to be printed through the todos filter before calling lpr. Instead of using lpr directly, you can now print with the command:

```
print your_text_file.txt
```

This is not ideal, but a very usable workaround. Do not use lpr directly if you have installed this workaround.

Alternative approach to printing in 2.11 BSD

Jon Brase has pointed out, that it is also possible to use the Raspberry Pi or a Linux laptop as a printer server, using cups-lpd.