September 1990 $2.50
Volume 2, Number 9

Technical UNIX® User Group

newsletter of the Technical UNIX® User Group

This month ...

Late Breaking News...
Next Meeting to be held at
UNISYS
See ANNOUNCEMENT for details

Thoughts From The Editor

By Susan Zuk

It's really sad to see summer coming to an end. It is such a short season. I was in Salt Lake City a few weeks ago and I was quite disappointed to see Christmas decorations in the stores. I think they are being a little too quick!!

Have you been involved in any exciting projects during the summer? Have you see any new technology or software? You should let the group know by writing a short article! Let us know what interesting things have been happening in Winnipeg.

This summer I was working on one of our newest UNIX computers. It is called the U6000/70 and U6000/80. The interesting point about these machines is that it has multprocessors. I was working on a U6000/80 with 20 processors, 64 MB of memory and 6 GB of disk. What an awesome machine. Now here's a machine I wouldn't mind having at home.

Our featured article is about remote spooling. It provides some helpful hints on how the System V spooler has been written to perform remote printing using uucp. This is a nice feature to have available if you have multiple machines at different locations and you need information printed from the remote sites but don't want perform a print screen or have users that need an easy to understand method for printing.

Gilbert also discusses what he has been doing this summer. He reminds us that the fiscal year is almost ended for the Technical UNIX User Group. How do you want to become involved this coming year? What can you contribute to the group and what do you want to see happen in the upcoming year? These are very important questions, since a group is only as good as its members.

Well, that's all for now and we'll see you at the meeting on September 11th.

Group Information

The Technical Unix User Group meets at 7:30 pm the second Tuesday of every month, except July and August. The newsletter is mailed to all paid up members 1 week prior to the meeting. Membership dues are $20 annually and are due at the October meeting. Membership dues are accepted by mail and dues for new members will be pro-rated accordingly.

The Executive

President:	Gilbert Detillieux	261-9146
Vice President:	Derek Hay	943-5401
Treasurer:	Gilles Detillieux	261-9146
Secretary:	Vacant	
Newsletter Editor:	Susan Zuk	(W) 788-7312
Membership Sec.:	Gilles Detillieux	(W) 788-6209
Information:	Gilbert Detillieux	261-9146
	(or) Susan Zuk	(W) 788-7312

Technical UNIX User Group
P.O. Box 130
Saint-Boniface, Manitoba
R2H 3B4

Copyright Policy and Disclaimer

The Technical UNIX User Group, the editor, and contributors of this newsletter do not assume any liability for any damages that may occur as a result of information published in this newsletter.

ANNOUNCEMENT...

Meeting Location:

The September meeting location will be held at the UNISYS Building, 10th Floor-1661 Portage Avenue.

The President's Corner

By Gilbert Detillieux

Well, we're back again, after what seemed like a very short summer break. Back for another year of meetings that will delve into the dark depths of the UNIX system and shed some light on its mysteries. This is a good time to reflect on where we've been, and where we are headed as a group.

This summer was an exciting time for us at the University. We recently got our connection to the Internet, that ad-hoc international network of computer systems that opens the door to some great new services, and also opens a Pandora's Box of potential security problems. So far, no security problems have really plagued us; possibly the conviction of Robert T. Morris was enough to discourage any other would-be crackers. So far, the Internet connection has been a great plus for us.

With the Internet, it is now possible to use Telnet to log into any other Internet system for which we have a valid userid. It is also possible, via a service called Anonymous FTP, to transfer files from archive sites everywhere. What sort of files? Well, archives of network news postings are available, providing a wealth of information, and a wealth of public domain software and other free software. We have already obtained, via FTP, lots of the latest GNU software (emacs, C, C++, Bison, and more), and many other interesting and useful utilities.

This free software, and the use of Anonymous FTP to acquire it, will be the focus of the September TUUG meeting, to be held at Unisys (10th floor, 1661 Portage) at 7:30PM, Sept. 11. Hope to see you there.

The Free Software Foundation, which provided the GNU software, has an interesting philosophy. It believes that software should be freely produced and distributed to all, to make computers more useful and make ourselves more productive. They encourage all users of GNU products to develop their own software and make it available by the same means.

While this altruistic goal may seem too idealistic for many, it is quite similar to the philosophy that allows computer user groups to succeed. Think about it: a user group's goal is to encourage the free exchange of information among its users, to make computers more useful and make ourselves more productive. Also, like the Free Software Foundation, a user group's goals can only be sustained if everyone involved is willing to take an active role and contribute something to the group. Those who have gotten something out of the group in the past should now be prepared to put something back in. It's only common sense.

On that note, it is with some regret (and also some relief) that I announce that I will not be running as president of the group for the next year, and encourage you to consider this position. Being president for the past two years (more if you count our early underground existence) has been a very interesting and worthwhile experience for me, and I don't regret one moment of it. With experience gained, and with other commitments on my time, it is time for me to move on. I still do intend to be available to the group in the capacity of past president, and active member.

I would also like to point out that in the past year, our executive shrunk from 6 at the start down to 3 remaining bodies - the same 3 who founded the group, and have been active in the executive from the beginning. It is time for some new blood and some new direction in this group. This will be essential to the group's continued survival. I now invite you all to consider what role you are willing to take, and what you are willing to put back into the club.

The Fortune File

Submitted by Gilles Detilieux

If you're not part of the solution, you're part of the precipitate.

UNIX Programming Tips

Remote Printing in System V

By Curt Motola & Bruce Stewart
Reprinted with permission from the September issue of CommUNIXations, published by Uniforum

UNIX System V includes a new print spooler that has significantly more flexibility than in earlier versions of the operating system. The spooler allows administrators to define multiple logical printers, each associated with a physical print device. A separate program controls the device interface so administrators can tailor print spooling to their own requirements. The flexibility of the System V print spooler can ease many of the problems of remote printing, as shown in the following interface script, which accepts remote print requests and routes them to the proper machine. Once this script is installed by the system administrator, users can route print requests to printers on other machines via the familiar 'lp' command. (This column assumes that machines are connected by a uucp network.)

When the user requests printing of a file via the lp command, the file is copied to the *spool* directory for the specified printer. Most of the lp options the user specifies are saved also. The file stays in this directory until the line printer daemon determines that the printer is available to print the file. The daemon executes the *interface* program (or shell script) and passes to the interface program the lp options that the user specified and the name of the file. The interface program handles the options (such as number of copies) and optionally may process the data, then delivers the data to the printer. For example, lowercase alphabetic characters could be shifted to uppercase for printers that don't print lowercase characters.

The UNIX command 'lpadmin' is used to define a new logical printer to the System V spooler. Each printer must be defined before it can be used. The definition establishes a correspondence between a logical device (the name of the printer), a physical device (the actual port or device special file) and the interface program that communicates between the two. Often, the interface program is a shell script that accepts input from stdin, formats or massages the text in some way, and writes the results to stdin to be taken from the file(s) to be printed and for stdout (and stderr) to be written to physical device associated with the logical printer.

To define a remote printer to the spooler, we, as root, use the command

```
lpadmin -premote -v/dev/null -irprint
```

This defines a logical printer named *remote* and associates it with the physical device /dev/null. Normally, the physical device for a printer would be a parallel or serial port on the machine. For remote printing, however, the physical device is attached not to the local machine but to another machine.

The local spooler has no knowledge of, or control over, remote physical devices. Instead, the interface program is used to connect the local (logical) printer with the physical device on a remote machine.

The -i option specifies the interface program that will control output to the physical printer. The interface program, by convention, has the same name as the logical printer to which output will be sent. The actual program lives in /usr/spool/lp/interface. Although the interface program may be a shell script or a compiled program, it must be executable by the lp user ID on the system.

For remote interface programs (such as this one), the file should also have execute permissions for the same group as the networking utilities used to communicate bewteen machines (in our case, uucp). Once the printer has been defined to the spooler, the enable and accept commands allow requests to be sent to the new printer.

The Interface Program

Our example (rprint) implements the interface program as a shell script. It assumes a uucp connection (direct or dial-up) between the machines that will share the printer. The script then saves the positional parameters that describe the job to be printed -- the logical printer name, the request ID, the user who is printing the job, the number of copies and any additional options passed to the interface program. Once the values of the first five positional parameters have been saved, the script uses the shift command to assign the filenames to be printed to $1, $2, etc.

The script now parses the options supplied when it was invoked. The options are of two types: those intended for the local spooler (the positional parameters) and those intended for the remote spooler. Options to the remote spooler must be preceded by "-o" on the command line. Any unrecognized options not preceded by "-o" are treated as errors. For example, the following command prints the file /usr/pub/ascii on a printer named *laser* on machine *norbert.*

lp -dremote -o"-m norbert" -o"-p laser" /usr/pub/ascii

The local spooler recognizes any options that begin with"-o" as ones intended for the local interface program. The local spooler removes the "-o" prefix before the interface program is invoked. The interface program uses the -p and -m options to route the print request. The -m option specifies the machine name, and the -p option gives the name of the remote printer. Any other options are assumed to apply to the remote printer. For these, the interface script appends a "-o" and passes them to the remote printer. This technique allows users on the local machine to specify options such as pitch, font and page orientation. The options are concatenated in the RMTOPTS variable. The script also recognizes standard options for the System V spooler (such as -c, -n, and -t) and formats them appropriately for the remote spooler.

Finally, the script verifies that a remote machine has been specified. If a remote printer has been specified, the script uses the "-d" option to identify the requested printer to the remote spooler. The script ends by invoking uux to transfer the local file to the remote print spooler.

The script shows one approach to solving the problem of remote printing. It is written to allow options to be passed to the remote spooler -- a useful feature if more than one printer is available on the remote machine or if printer-specific options must be specified. This approach also allows one interface file to direct print requests to multiple machines and printers, since both the remote printer and remote machine names may be passed to the local interface file as parameters. However, the script owes its real flexibility to the System V spooler, which allows a separate program to control the interface between a logical printer name and a physical print device.

rprint

```
#
# Sample model file for remote printing
#
PATH=/bin:/usr/bin
REMOTELOG=/usr/spool/lp/remote.log
RMTOPTS=
MACHINE=
SPOOLER=lp
#
# set positional parameters -- NOTE that some systems
#             may use a different number of parameters
PRINTER=`basename $0`
REQUEST=$1
NAME=$2
TITLE=$3
COPIES=$4
OPTIONS=$5
shift 5       # move the filename(s) to the head of the list
FILES=$*
# now parse the options, some of which may be for the
# remote printer.  A -m option precedes the remote
# machine name; the -p option identifies the printer on the
# remote machine.  Any other options are passed on to
# the remote printer.
#
set -- $OPTIONS
while [ ! -z "$1" -a "$*" != "$FILES" ]
do
      case $1 in
      -c|c) RMTOPTS="$RMTOPTS $1";       shift;;
      -m|m) MACHINE=$2;                  shift 2;;
      -n|n) RMTOPTS="$RMTOPTS $1$2";     shift 2;;
      -p|p) RMTPRINT=$2;                 shift 2;;
      -r|r) RMTOPTS="$RMTOPTS $1";       shift;;
      -s|s) RMTOPTS="$RMTOPTS $1";       shift;;
      -w|w) RMTOPTS="$RMTOPTS $1";       shift;;
      -t|t) RMTOPTS="$RMTOPTS $1$2";     shift 2;;
      *)    RMTOPTS="$RMTOPTS -o$1";     shift;;
      esac
done

# Check for machine name ... we have to know where to
# send
if [ -z "$MACHINE" ]
then
      echo "No remote machine name specified .. cannot prt" \
            >> $REMOTELOG
      exit 1
fi
# Check for remote printer specified, and format properly
#
if [ ! -z "$RMTPRINT"  ]
then
      DEST="-d$RMTPRINT"
fi
# We're ready to print. Add entry to remote print log
(while [ "$COPIES" -gt 0 ]
do
  for FILE in $FILES
      do
            cat "$FILE" 2>&1
            echo "`date+%D-%T`:$PRINTER:\
            $MACHINE:-t$TITLE:\$OPTIONS:\
            -n$COPIES:$FILES" >> $REMOTELOG
      done
      COPIES=`expr $COPIES - 1`
done
) | uux - $MACHINE!$SPOOLER "$DEST" "$RMTOPTS"
```

Curt Motola *is president of Classic Software, Inc., a supplier of UNIX consulting services.*

Bruce Stewart *is a computer consultant experienced in program development, systems administration and system design.*

Technical UNIX® User Group

Agenda

for

Tuesday, September 11th, 1990
7:30pm
UNISYS Building
10th Floor-1661 Portage Avenue

1. Round Table	7:30
2. Business Meeting	8:00
a) Membership Secretary's Report	
b) Newsletter Report	
c) Treasurer's Report	
4. Break	8:30
5. Presented Topic	8:40
Internet Freeware	
6. Adjourn	9:30