December 1988
Volume 1, Number 3

Technical UNIX® Users Group

newsletter of the Technical UNIX® User Group

This month ...

UNIX is a registered trademark of AT&T.

Editor's Note

by Darren Besler, Editor

Welcome to the third newsletter of the Technical UNIX User Group. It's amazing how sometimes one has a lot to say and then other times one can't think of anything to say at all. Well, this is one of those times for me. I really can't think of much to say. I think I must be working too hard or possibly not hard enough!

Anyhow, this month we have an article describing a shell program (source included) for adding a new userid to a UNIX system. Even if you have administrator utilities for adding new users this article and program is well worth reading for the educational merit alone. Also, our first non-executive contribution is some information on a tape backup system from PCS Technologies. Thank you Ken Wilkie for your contribution. I had another article on the Internet Worm but this issue has become big enough already so it will have to wait till next month.

I would like to thank Gilles Detillieux, Susan Zuk, Gilbert Detillieux, and Ken Wilke for their contributions to this months newsletter. Again, thanks goes out to UNISYS for their assistance in producing this newsletter.

Well, I really can't think of much else to say other than have a Merry Christmas and I hope to see you at the next meeting.

The Executive

President:	Gilbert Detillieux	261-9146
Vice President:	Vacant	
Treasurer:	Gilles Detillieux	261-9146
Secretary:	Susan Zuk	(W) 786-8483
Newletter Editor:	Darren Besler	(W) 934-5475 (H) 254-3392
Membership Sec.:	Pat Macdonald	(W) 474-9870
Ilnformation:	Gilbert Detillieux Susan Zuk	261-9146 (W) 786-8483

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

Meeting Schedule

2nd Tueday of every month
7:30 PM
Room 431,
Basic Medical Sciences Building
Health Sciences Center

Copyright Policy and Disclaimor

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

President's Corner

by Gilbert Detillieux, President

The convention center floor was packed, as thousands of attendees slowly worked their way through the aisles. Row after row of exhibitors, anxious to show their equipment to anyone who paused long enough to look interested. Sun workstations here, VAX workstations there; small PC and Mac based systems, and high power super-micros by Silicon Graphics; even a booth using Masscomp gear. High-resolution color displays as far as the eye can see.

Where was all this high tech UNIX equipment to be found? COMDEX? UNIX/Expo? Would you believe the Society for Neuroscience's annual meeting. That's where your humble president spent a week in November, along with 12,000 other attendees who took over the Metro Toronto Convention Center (all three floors) for their week-long meeting.

The exhibit area was quite a sight. Along with all the publishers of medical texts and vendors of lab equipment, there were a whole bunch of companies selling computer systems, many UNIX-based, for applications such as analog data acquisition and analysis (one of my areas of interest), image analysis, and 3-D image reconstruction. This last application saw the greatest increase, and the greatest competition. At last year's show, there were maybe two or three companies with such systems; this time there were at least six.

All in all, there was more interesting computer hardware and applications than I've ever seen at any of the computer shows I've attended here in Winnipeg, in Montreal, or in Toronto.

There were a lot of PC based systems and some Macintosh based systems as well, mostly for less demanding signal processing applications. One company even had an Amiga based system, cleverly concealed behind a front panel of their own design. It's curious to see the stigma that seems to be attached to this computer, which is actually better suited to this application than a PC.

One machine that was not there, but I predict will be featured prominently at next year's show, is the NeXT workstation. The machine is almost ideal for that market: academic types, with limited budgets, requiring lots of computing power, and good graphics, particularly for applications such as signal processing. (Hmm... Let's see, how long would it take to port our software from the Masscomp?...)

Meetings, meetings, meetings. On to the next one...

Our December meeting will be back at the usual place and time (7:30 PM, Tuesday December 13, at 431 Basic Medical Sciences Building). For those of you who have never been there before, the entrance is at 770 Bannatyne (the Medical College), which is on the south side of Bannatyne, just west of Emily Street. Once you find the entrance, there are signs posted to guide you to the meeting room.

This meeting will be an informal get-together, as we are getting close to Christmas, and will be a good opportunity for new members to get to know others in the group. No business this time around, just lots of good cheer!

For the January meeting, we'll probably be back at Unisys, to continue the system administration workshop. (Did anyone have trouble finding it last time?) This time around, we should have all the technical glitches with the terminal emulation/overhead projector setup ironed out, so it should go more smoothly than last time. There was a very lively discussion going on at the November meeting, which was very nice to see. We unfortunately didn't cover nearly as much material as we hoped, but we can make up for that at future meetings, as long as everyone is still interested.

• •

Elsewhere in the newsletter, there is a nifty little shell program to automate the process of creating new users on a UNIX system - very useful on systems without system administration menus. Since most of the November workshop dealt with this subject, this program may come in handy for a lot of you out there.

This is also the first newsletter to feature a submission from a group member other than the executive. Thanks, and keep the material coming; we sure can use it!

Speaking of newsletters, just a reminder that those of you who don't sign up as members soon may find yourselves cut off from our mailing list. The January newsletter is the last one that will be sent to non-members! Avoid disappointment, and sign up either at our December meeting, or get in touch with our membership secretary, Pat Macdonald, or any member of the executive for that matter.

Best wishes to all for a very merry Christmas, and I hope to see
you all at our meetings in the New Year!

The "newuser" program

By Gilles Detillieux

One of the first duties of any UNIX system administrator is to set up user IDs for new users. Although this is not a complicated task, it does get a little tedious. Also, because it involves several steps, it's easy to forget a step now and then.

Some UNIX systems provide some way of automating this process, often in the form of a menu-driven system administration program. Unfortunately, with most UNIX systems you're on your own.

The solution I adopted, for the MASSCOMP machines I look after, was to write a shell procedure which handles all of the tedious work involved in user ID creation. It simply prompts me for the necessary information for each user, then creates the user's home directory, sets up ownership and permissions, and creates the new entry for the password file. The program, written for the Bourne shell, is given in the listing below. Although written for the MASSCOMP, it is general enough that it should work on any UNIX system.

The first thing the program does is set variables for various parameters and initial default values. The password file lock name, defined on line 7, is used to prevent conflicts with the "passwd" command. When the "passwd" command modifies the password file, it creates this lock file, so that if another user uses the "passwd" command at the same time, he will be locked out until the first one has finished modifying the password file. This program uses the lock file a similar way.

Lines 9 to 15 define all of the fields in a password file entry. Most are just set to an empty string here, and are set later. Others are set to their initial default value. These can be changed, as appropriate, for your own system.

The "homepath" variable, on line 17, defines the directory under which the users' home directories will be created. This varies from system to system, but is usually something like "/usr", "/u", "/usr2", "/usr/acct" or "/mnt". The "sample" and "csample" variables define the home directories of a Bourne shell user and a C shell user, respectively, from which we can copy ".profile" or ".login" files for new users. Set these to names appropriate for your system.

Lines 21 to 25 check to make sure that the person running the program has write-permission on /etc/passwd. This will prevent anyone but the super-user from running the program. (Your password file can only be written by "root", right?)

The rest of the program is one large loop, which prompts for and reads the login name for the next user to be created. The loop terminates when end-of-file is entered. Lines 30 to 32 also check for other exit conditions. If an empty line is entered, or if any of the words "quit", "exit", "q", "e", or "x" are entered in upper- or lower-case, the program terminates.

Line 34 tests whether the given login name is already in the password file (as the first field in a line). If it is, the program returns to the start of the loop, reprompting you for a name.

Line 40 simply sets the "homedir" variable, for later use, by appending the new login name to the default parent directory name.

Line 44 shows the numeric user IDs in use in /etc/passwd. Fields 1 and 3 (login name and numeric UID) are cut out, and displayed in 5 columns. The sed command deletes any blank lines. The same operation is performed on /etc/group, on line 59. Line 47 finds the maximum numeric UID in use; the next number will be the default value for this user's UID.

Lines 50 to 55 prompt for and read the UID. If end-of-file is entered, the program terminates. Otherwise, the variable UID is set to the value entered, as long as it isn't simply a blank line. If a blank line is entered, the variable keeps its current value. The same approach is used for prompting for all of the other fields.

Once all fields have been entered, the program pieces together the password file entry (line 93), and shows it. Note that the fifth field (the comment field) is used only for the user's full name. If your system requires other information here, you will have to make the necessary changes to this code. The program then asks for confirmation. If a blank line or a line beginning with a "y" is entered, the program goes ahead (line 100). If the line begins with something else (line 101), then the program restarts the main loop, reprompting you for all fields, as before. The GID and full name fields will retain the value you last entered. Again, an end-of-file will cause the program to terminate.

Lines 108 to 117 will test for the existence of the user's home directory. If it doesn't exist, it attempts to create one. If this fails, the program restarts the main loop, reprompting you for all fields.

Lines 120 to 137 make four attempts at appending the new entry to the password file. Each time, the program tests for the lock file. If it's there, the program pauses for three seconds before trying again. If it isn't there, it creates it, writes the password file entry, then removes the lock. Once this is done, it clears the loop index, to indicate success, then breaks out of this loop. If all four attempts fail, the loop index will have the value "0", causing the program to restart the main loop (lines 134 to 137).

Lines 140 and 141 assign the user's UID and GID to his home directory, as though he had created it himself. Line 142 sets the mode of the home directory, giving him and his group write permission, and giving all users read and search permission. If this mode is not appropriate for your system, change this line.

If the last part of the home directory name is the user's login name, line 146 will extract the parent directory name, and keep this as the default value for the next time through the loop. The full name field variable is then cleared, so no default value exists for the next pass through the loop.

The rest of the code in the loop handles copying of shell start-up files. If the login shell field is empty or ends with "sh", then code appropriate for the Bourne shell is executed. If the field ends with "csh", then code appropriate for the C shell is executed. In both cases, the program asks whether the file(s) are to be copied. If the response is a blank line, or begins with a "y", it asks for the name of the directory from which to copy the file(s). It then copies the files, and changes their ownership and permissions.

After all that, the program goes back and asks for the next user's login name. Simple enough?

Good luck in getting this working on your system. If you want to try it out, but are worried about clobbering your password file, you can test it by changing the "pw" variable to some other value, such as "/tmp/passwd". Then copy the password file to the file named above. When you test out the program, only this file will be changed, not the real password file. (It will also still create home directories, so you'll have to remove these after testing.) This is a good way of trying out any changes you make to the program, or even just to test it after you type it in.

I'd be interested in hearing from anyone who runs into problems running this on their system, or hearing of any suggestions for enhancements to the program.

```
Listing:  The "newuser" program.

     1   #! /bin/sh
     2   #
     3   # newuser - set up passord file entries and home directories for new users
     4   #
     5
     6   pw=/etc/passwd        # password file
     7   pwlock=/etc/ptmp      # password file lock, used by "passwd" command
     8
     9   logname=              # user's login name, read from standard input
    10   pword=                # encrypted password field left blank by default
    11   UID=                  # numeric user ID.  default value calculated later
    12   GID=100               # default numeric group ID for new users
    13   fname=                # user's full name, read from standard input
    14   homedir=              # user's home directory name, default value set later
    15   logshell=/bin/csh     # default login shell for new users
    16
    17   homepath=/usr         # default parent directory of users' home directories
    18   sample=/usr/fred      # location of sample Bourne shell ".profile"
    19   csample=/usr/wilma    # location of sample C shell ".login", etc.
    20
    21   if [ ! -w $pw ]
    22   then
    23       echo "$0: can't write to $pw, need to be super-user" >&2
    24       exit 1
    25   fi
    26
    27   while echo "\nUser's login name:      \c" && read logname
    28   do
    29       # Control-D, newline, "quit", "exit", "q", "e", or "x" will end loop
    30       case "$logname" in
    31       ""|[Qq][Uu][Ii][Tt]|[Ee][Xx][Ii][Tt]|[QqEeXx])  break ;;
    32       esac
    33
    34       if grep -s "^$logname:" $pw
    35       then
```

```
        echo "$0: login name $logname already in $pw" >&2
        continue
    fi

    homedir="$homepath/$logname"    # default home directory

    # Show numeric user IDs in use.
    echo "UIDs in use:"
    cut -d: -f1,3 $pw | pr -t -w80 -o5 -5 | sed '/^ *$/d'

    # Find largest UID & add 1 to it, to get default for new UID.
    LASTUID="`cut -d: -f3 $pw | sort -nr | line`"
    UID=`expr $LASTUID + 1`
    echo "User's numeric UID:     [$UID] \c"
    if read x
    then
        UID="${x:-$UID}"
    else
        break
    fi

    # Show group IDs defined in /etc/group, ask for group ID.
    echo "GIDs defined:"
    cut -d: -f1,3 /etc/group | pr -t -w80 -o5 -5 | sed '/^ *$/d'
    echo "User's numeric GID:     [$GID] \c"
    if read x
    then
        GID="${x:-$GID}"
    else
        break
    fi

    echo "User's full name:       [$fname] \c"
    if read x
    then
        fname="${x:-$fname}"
    else
        break
    fi

    echo "User's home directory:  [$homedir] \c"
    if read x
    then
        homedir="${x:-$homedir}"
    else
        break
    fi

    echo "User's login shell:     [$logshell] \c"
    if read x
    then
        logshell="${x:-$logshell}"
    else
        break
    fi

    # Piece together password file entry, show it, and get confirmation.
    pwline="$logname:$pword:$UID:$GID:$fname:$homedir:$logshell"
    echo "\n$pwline"

    echo "Is this correct? (Y/N): [Y] \c"
    if read x
```

```
        else
            break
        fi
        echo “Copy from which directory? [$sample] \c”
        if read x
        then
            sample=“${x:-$sample}”
        else
            break
        fi
        cp $sample/.profile $homedir
        chown $UID $homedir/.profile
        chgrp $GID $homedir/.profile
        chmod 0644 $homedir/.profile
        ;;

    */csh)      echo “Copy .login, .logout & .cshrc” \
                     “from another directory? (Y/N): [Y] \c”
        if read x
        then
            case “$x” in
            “”|[Yy]*)   ;;
            *)          continue ;;
            esac
        else
            break
        fi
        echo “Copy from which directory? [$csample] \c”
        if read x
        then
            csample=”${x:-$csample}”
        else
            break
        fi
        cp $csample/.login $csample/.logout $csample/.cshrc $homedir
        chown $UID $homedir/.login $homedir/.logout $homedir/.cshrc
        chgrp $GID $homedir/.login $homedir/.logout $homedir/.cshrc
        chmod 0644 $homedir/.login $homedir/.logout $homedir/.cshrc
        ;;
    esac
done
```

PCS 2100 Tape Backup System

Contributed by Ken Wilke

[ed. The following is some information passed on to me from Ken Wilke of Price Waterhouse.]

Following you will find some information on the tape backup system from PCS Technologies Inc. This unit is now working at my clients installation and is very impressive. This unit can backup and verify 129920 blocks in 20 minutes and all files backed up can be selectively restored.

One of the great advantages of this unit is an unmanned backup can be performed, under tight controls, in multi user mode. We are still getting the bugs out of our backup but we hope to have it work to backup 3 drives at 129920 blocks and 1 at 69538 every night. This takes about 1 hour and 20 minutes.

Bob Norris of NCR is our contact for this device. His phone number is 452-3243.

```
then
    case “$x” in
    “”|[Yy]*)   ;;
    *)          continue ;;
    esac
else
    break
fi

# Test for existence of home directory, make it if it’s not there.
if [ ! -d $homedir ]
then
    if mkdir $homedir
    then
        :
    else
        echo “$0: User $logname not created” >&2
        continue
    fi
fi

# Append new entry to password file, as long as it’s not locked.
for tries in 3 2 1 0
do
    if [ ! -f $pwlock ]
    then
        : > $pwlock
        echo “$pwline” >> $pw; rm -f $pwlock
        tries=
        break
    fi
    case “$tries” in
    0)  ;;
    *)  sleep 3 ;;
    esac
done
case “$tries” in
0)  echo “$0: $pw is locked, try later.” >&2
    continue ;;
esac

# Set up ownership, group, and permissions properly for home directory.
chown $UID $homedir
chgrp $GID $homedir
chmod 0775 $homedir

# Extract parent directory name where appropriate, in case it’s changed.
case “$homedir” in
*/$logname) homepath=”‘expr $homedir : ‘\(.*\)’/$logname‘“ ;;
esac

fname=          # default full name cleared for next time

# Optionally copy startup files for Bourne shell or C shell.
case “$logshell” in
*/sh|””)    echo “Copy .profile from another directorv? (Y/N): [Y] \c”
    if read x
    then
        case “$x” in
        “”|[Yy]*)   ;;
        *)          continue ;;
        esac
```

PCS 2100 Product Description and Specifications

Product Availability: Including daisy chain cable for simple top of system installation for:

NCR 32 Series
UNISYS 5000 Series
ARETE - All Models
Others to be Announced

Prerequisite: SCSI Interface.

Maintenance: By PCS at $500 annual cost (90 day warranty) for overnight replacement of complete unit.

Availability: 30 day delivery.

Reliability: Non-recoverable error rate - Less than 1 in 10,000,000,000,000 bits read. MTBF: Greater than 15,000 hours at typical use.

Performance: 10.3 to 15 MB per minute (depending on system model) copied to standard 8mm cartridge tape. Unattended capacity - 2.1 GB (2100 MB) formatted on one tape.

Media: Standard 8mm tape cartridge (3.7" x 2.5" x 6"). Storage space is 1/100th of comparable 9 track tape.

Physical Dimensions: 7" width x 20" depth x 4.5" height.

Engineering: Safety specifications for UL, CSA, and VDE Emmission specifications for FCC, VDE, Class B.

Power: 110 or 220 volts at 50 or 60 Hz and 60 watts.

Environmental: Operating temperature: +5 degrees (C) to +45 degrees (C). Non-Operating temperature: -20 degress (C) to +60 degress (C). Relative Humidity (non-condensing): 20% to 80%.

Format

Recording Format: Helical Scan
Head Format: Read after write with separate full width erase head
Linear Recording Density: 43200 BPI
Track Density: 819 TPI
Areal Recording Density: 35 Million bits/sq. in.

Tape Motion

Tape Speed: 0.429 ips/10.89 mm/sec
Rotor Speed: 1800 RPM
Effective Head to Tape speed: 150 IPS
Rewind Speed: 8 times nominal speed

Controller Features

Integrated SCSI Controller and Formatter
Standard SCSI Interface Connection
Onboard Error Correction Code (ECC) and Error Recovery Procedures (ERP)
ECC: Reed-Solomon Product Code
256 KByte Speed-Matching Buffer

SCSI Characteristics

SCSI Specification - ANSI Version 17B
Conformance Level - 2
Sequential Access Devices
SCSI Bus Parity - Configurable (MODE Select)
Full Disconnect, Arbitration, Reconnect
Asynchronous Data Transfer Supported
Differential
No. Connecters - 1
Termination - Install/De-Install
Termination PWR - Use/Pass-Non Supply

SCSI Command Set

Test Unit Ready
Rewind
Request Sense (extended)
Read Block Limits
Read
Write
Write Filemarks
Inquiry
Space
Recover Buffered Data
Mode Select
Reserve Unit
Release Unit
Copy (* per User's Requirements)
Erase
Mode Sense
Load/Unload
Receive Diagnostic Results
Send Diagnostics
Prevent/Allow Medium Removal

Minutes From the Business Meeting
November 8, 1988

1. Minutes:

MOVED: (Susan Zuk) The minutes from the October 11th meeting be approved.

SECONDED: (Darren Besler)

In Favour: 12	Opposed: 0	Carried

2. Membership Dues (Corporate vs Individual Membership Fees):

Discussion was brought to the table to have corporate rather than individual memberships. It was decided that it is easier to keep control if there are individual memeberships. It was also stated that only one newsletter is available per membership so there is a chance that information might not be circulated within the organization before the upcoming meeting occurs.

Ammendment to the Motion (On the meeting of the 2nd week of October, a membership fee of $20.00 will be charged. Members joining later will pay a pro-rated amount. Membership fees will not be refundable.)

Ammendment: (Gilbert Detillieux) The only category for membership is an individual membership fee.

Seconded: (Peter Somers)

In Favour: 12	Opposed: 0	Carried

3. Newsletters:

A request was made for members to submit articles for publication in the newsletter.

4. Membership:

At the present time we have 9 paid members and $180.00 in our bank account. Please pay your fees soon.

At this time a portion of collected fees have paid for cheques being printed, newsletter creation, postage, copy paper and registering our UNIX group name. Non-members will receive only two more newsletters (the December and January issues).

by way [illegible] name retention

The group name TUUG (Technical UNIX User Group) has been registered ~~as a non-profit group and is registered~~ for three years at which time it is renewable.

[illegible]

3. Voting on our ~~Official Group Name~~:

newsletter "the "newsletter of the

MOTION: (Darren Besler) The official name of the ~~UNIX group~~ will be ~~TUUG (~~Technical UNIX User Group".

SECONDED: (Gilbert Detillieux)

In Favour: 12	Opposed: 0	Carried

DECEMBER AGENDA

Christmas Get Together