March 1991 $2.50
Volume 3, Number 6

Technical UNIX User Group

newsletter of the

Technical UNIX® User Group

This month ...

Late Breaking News...
Next Meeting to be held at the
XCAN GRAIN LTD.
See ANNOUNCEMENT for details

Thoughts From The Editor

By Susan Zuk

This month's newsletter includes a handy shell script submitted by Kirk Marat. What this shell script does is allow the user to change the user and group id's recursively down a directory tree. I can think of many times when a script of this sort could have come in handy.

Our feature article deals with ways of handling your filing system. It discusses the tree structure of UNIX, how to find specific information in a group of files, and how to remove files. This is a good article for those users just starting to work with UNIX.

The final article is a review on two books for MS-DOS users wanting to learn how UNIX compares with DOS. They look to be quite a good investment for those DOS gurus wanting to delve into the world of UNIX. Read the article and see if it might be a good buy for you or one of your fellow workers.

Two sessions about and relating to UNIX are going to be held in Winnipeg in April. Both are being sponsored by CIPS. One is the April 16th Dinner Meeting entitled "The Appeal of UNIX" and the other is a Seminar which CIPS will be holding on April 9th. The seminar is called Downsizing With Client/ Server Approaches. What this seminar will discuss is how technology is changing the workplace by using small machines, such as PC's, workstations, LANS, client/server DBMS and open operating systems, to surround and sometimes replace mainframes. If you are interested give Al a call at 986-2143.

Well, that's all I have for now. See you at the meeting!

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:	Eric Carsted	1-883-2570
Past President:	Gilbert Detillieux	261-9146
Treasurer:	Gilles Detillieux	261-9146
Secretary:	Vacant	
Newsletter Editor:	Susan Zuk	(W) 788-7312
Membership Sec.:	Darren Sampson	(W) 783-4614
Meeting Coordinator	Kirk Marat	(W) 474-6259
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

This newsletter is ©opyrighted by the Technical UNIX User Group. Articles may be reprinted without permission as long as the original author and the Technical UNIX User Group are given credit.

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 March meeting location will be provided by Xcan Grain. Xcan is located in the Toronto Dominion Building at 1200-201 Portage Avenue. Please enter at the front door. Eric Carsted or the security guard will be ready to allow you access to the elevator. If the security guard asks, say that you are with TUUG. See you at the meeting.

President's Corner

By Eric Carsted

If you've read a Unix rag in the last month or so or even one of the PC magazines you would have read that "OS/2 is dead" or something to that effect. This coming about as result of a fallout between IBM and Microsoft. The general impression was that the success of Microsoft Windows 3.0 revitalized DOS and with its memory management and task switching capabilities OS/2 would not be needed by the majority of PC users. Other factors that slowed acceptance of OS/2 were the requirements of large amounts of memory and CPU power coupled with a lack of applications that used the added functionality of OS/2 and a lack of printer drivers.

Some of the articles speculated that OS/2 would be back and that it would kill Unix. Others felt that this was a great victory for "Open Systems" and Unix because if the two largest operating system developers, IBM and Microsoft, could not introduce a new proprietary operating system no one could. I feel that both sentiments could be correct. I don't think that Microsoft is going to concede the battlefield to Unix. A multitasking, multiuser operating system, with the Windows interface, running Windows applications without modification, and good networking capabilities will take over the current DOS market. On the other hand I don't think that this is going to be available for at least two years and probably closer to four. I think that Microsoft by cutting its ties to IBM will allow for input into this operating system to come from both the user community and software developers. This will remove a lot of the criticism that OS/2 received in the media and make it more appealing to the end user. On the hardware requirement front the average system going out the door is a 386/SX with 2MB RAM which should be able to run this new OS with just the addition of a few Megabytes of RAM. By the time this new OS is released it should be a solid, well tested product with a large hardware base in the market ready to run it.

What does this all mean to the Unix community? I think it means that Unix has a window of opportunity to assert itself as a solution today for those wanting those features in OS now. Unix has to prove to the DOS community that it does not require a "Unix Guru" to administer it. Unix has to show a large base of applications. Unix has to provide one standard GUI for application developers and users. Unix has to demonstrate its scalable path for growth both in networking and multiprocessing. Unix has to prove it is a cost effective solution for small organizations as well as large corporations. Unix has to stop cutting its own throat by all the infighting between UI and OSF.

I think a good start in this direction is The Santa Cruz Operation's ODT or Open Desk Top. ODT bundles SCO Unix 3.2.2, TCP/IP, LAN Manager, NFS, the Ingres Database, X windows, Motif, Merge 386 (DOS under Unix), and PC/Interface in a two user package for around $900.00 and can run on hardware costing as little as $3,000.00. The installation is relatively painless and administration can all be done through a menu driven shell. The server upgrade provides for unlimited users and expanded NFS capabilities at around $1,200.00 and a full developers kit with X windows and Motif libraries for another $1,200.00. This combination of applications and Unix allows the user to save all his DOS programs and enter the world of Unix. I will be giving a demonstration of it at the next TUUG meeting.

UNIX For Beginners

By Dave Taylor

Reprinted from /usr/group CommUNIXations February1991

The UNIX File Structure

The hierarchical file system is a fundamental feature of the UNIX operting system. When properly understood, it can help you organize files, find information faster and work more efficiently. The UNIX file system is organized as a "tree" of entries, with each entry being either a branch (a directory) or leaf (such as a file or a device). The top of the tree is the / directory and beneath it sub-trees can be on different disks, different computers (via hook-up using the Network File System or a similar structure) or even completely different media (such as optical disks). Almost all operating systems have tree-style file systems.

Organizing a large number of items that resemble each other is an inherently difficult task. The logical way to divide them is by their function. In UNIX, for example, device drivers are found in /dev, temporary files in /tmp, spooled files in /usr/spool and user home directories in places such as /users or / home. Not only does this organize the thousands of files found on a UNIX system into manageable subsets, it enables commands that read through a specific directory looking for files to work faster, since there are fewer entries to search through in each directory.

Learning how to create subdirectories, organize files and manoeuvre within the subtree of your home directory can help any user streamline many tasks. For example, imagine that you've just logged in and typed **ls** to see the files in your home directory. The output, shown in Listing 1, is a jumble of different files. You may not even remember the contents of some of them.

Listing 1. A disorganized home directory

```
$ ls
FOO           memo.1      report         test
NOTES         memo.2      resume         test.o
calendar      memo.3      save           test.c
dead.letter   memo.4      saved          to.mom
demo          memo.5      saved.mail     to.sondra
mailbox       misc        save.mailbox   xx
mbox          news.info   stuff
```

Finding information in this mess is difficult but you can discern a few main partitions into which the files may be divided. To accomplish that, use the mkdir command to make new subdirectories (such as mkdir Personal) that correspond to your primary functions or projects. For example, you might have a yourname/lib directory for "boilerplate" files and a yourname/memo directory for memos. After that, use the mv command to move files into the different subdirectories (mv to.mom Personal) and the cd command to move to the directory of choice (cd Personal). You might also consider establishing a set of basic directories that your co-workers could use.

The ls command is the best option for listing files in a specific directory, but adding some parameters can make the output even more useful. The first one is -F, which causes **ls** to indicate the type of the file through a trailing ASCII character. Typically, / indicates a directory and * an executable file. The C option can be added, as in **ls -FC**, which lists files in multiple columns. (Most Berkeley UNIX systems alias the **ls** command to provide columnar listings, as shown in these examples.) Another useful flag is -R, which requests that the listing be of the entire file tree, starting at the current directory. Listing 2 shows such an organization, using both the -F and -R flags.

Finding Files

Operating systems don't naturally make it easy to find specific text in a group of files. Establishing naming conventions for yourself will help to make those names more memorable. You can also use the UNIX grep command (global regular expression print), which searches through a set of files for a specific pattern. For example, to look for the file that contains the word birthday, type **grep birthday *** within your home directory. The output would list all lines (with their file names) containing that pattern within the specified set of files.

In a hierarchical directory structure, however, you can't use this simple grep pattern * command because the * won't expand to include files within subdirectories. However, the command shell has ample power to deal with this situation. The solution is to use instead the command grep pattern */* to look at the files in all your subdirectories. There's a subtle catch to that command, though; */* expands to all files within subdirectories but not to files in the current directory. To include that, you must use * as well as */*, leading to the final search command of grep pattern * */*.

REMOVING FILES

When you want to remove all the files in a subdirectory, you might also want to remove the directory itself. The rmdir command serves this purpose. Unlike some UNIX commands, rmdir is rather graceful about ensuring that you won't accidentally delete important files. For example, if you mean to remove the directory tmp but mistakenly type **rmdir tom**, rather than removing the directory and all files within it, **rmdir** would respond, rmdir: tom not empty and fail. To empty a directory, use the cd command to get into it and **rm** to remove the files. Make sure you don't leave any hidden files, by checking with **ls -a**.

```
Listing 2. The same directory, organized

$ ls -FR

FOO        Personal/       demo          save
Mail/      Programming/    misc          saved
Memos/     calendar        news.info     stuff
NOTES      dead.letter     report        xx

./Mail:
mailbox    mbox            save.mailbox  saved.mail

./Memos:
memo.1     memo.3          memo.4        memo.5
memo.2

./Personal:
resume     to.mom          to.sondra

./Programming:
test                       test.c        test.o
```

Wild cards such as * also work with the rm command. It is safer, though, to use the "interactive" mode of rm with the -i option, as in **rm -i ***. The system then prompts you for each file name generated from the * wild card.

Organizing your files to take advantage of the UNIX hierarchical file system is just one example of a number of different ways that you can exploit the basic design of the operating system to improve your organization and productivity. But like any other productivity tool, it works well only if you take the time to getfamiliar with it.

Dave Taylor is president of Intuitive Systems, a consulting firm in Los Altos, CA, which specializes in internationalization, user interface design and software marketing. He is a contributing editor of CommUNIXations.

A Shell Script to Recursively Change Owner & Group ID's

Submitted By Kirk Marat

```
# touser changes the user and group id's
# recursively down a directory tree
# Source:
#  Bruker Analytische Messtechnik GmbH
#  D-7512 Rheinstetten, FRG
#

if [ $# -lt 2 ]
then
  echo Use: touser user dirname
  exit 1
fi

user=`grep “^$1:” /etc/passwd | line | cut -d: -f3`
group=`grep “^$1:” /etc/passwd | line | cut -d: -f4`
shift
find $* -exec chown $user {} \; -exec chgrp $group {} \;
```

BOOK WATCH

By Stephen M. Chan

Reprinted from /usr/group CommUNIXations March1991

UNIX GUIDE FOR DOS USERS
Allen Taylor
MIS Press; 230 pages, $19.95US

INFOWORLD: A DOS USER'S GUIDE TO UNIX
Douglas Topham
Brady/Simon & Schuster; 273 pages, $27.95US

These books are designed to help experienced MS-DOS users make the transition to UNIX systems. Both are written in an informal (as opposed to textbook) style, are easy to read and not excessively lengthy or overly complex.

Taylor's book is aimed at DOS users rather than DOS programmers and does not assume knowledge of programming languages. His approach does presume that anyone who knows MS-DOS can master UNIX rather quickly. He demonstrates this by building on the reader's familiarity with DOS and making comparisons between DOS and UNIX commands.

UNIX Guide for DOS Users gives an overview of UNIX and describes its structure in relation to that of DOS. Its other chapters cover hardware requirements, disk usage and management, basic UNIX operations, file operations, text processing, multiuser operations, process management, communications, shell programming, systems administration, running DOS as a subtask under UNIX, printing, text formatting, software development and the X Window System.

Taylor begins with the areas of similarity between the two operating systems and then moves into areas of UNIX that have no DOS equivalents. Since UNIX was conceived as a software development environment, it offers numerous tools that are available in writing applications programs. Late in the book, Taylor provides an appreciation for these programming tools anfd offers insights for those readers who wish to take advantage of them, but his focus overall remains on the major facilities.

A DOS User's Guide to UNIX shows the direct correlations between DOS and UNIX from a DOS user's perspective, pointing out that many DOS features were taken directly from UNIX. Topham, a technical writer for *InfoWorld*, emphasizes that it is not necessary to give up DOS to use UNIX and that users can improve their productivity by using both operating systems. He correctly notes that MS-DOS provides computational independence at an economical cost, while UNIX can supply cohesiveness via a company or departmental database and electronic mail system.

Like Taylor, Topham begins by comparing DOS and UNIX, pointing out both UNIX's strengths and weaknesses. He moves on to basic UNIX procedures, such as logging in and changing passwords. Succeeding chapters cover files and directories; command processing; utility programs; basic editing; text formatting; communcations; DOS batch versus UNIX initialization files; I/O redirection; Bourne versus C shells; system administration; managing disks, tapes, terminals and printers; and DOS versus UNIX startup and shutdown procedures.

Topham makes constant comparisons between the two operating systems, effectively anchoring the reader on familiar ground while covering new concepts. His extensive use of computer sessions (printed in different fonts to help separate text, computer-generated output and user-typed commands) along with tables and diagrams make this book a model of clarity. For example, in coverage of the UNIX *chmod* command, Topham uses tables to display UNIX's octal arithmetic calculations, which are required to convert a *chmod 754 filename* to a -rwxr-xr file access permission. In addition, there is an appendix of UNIX commands and command options, along with a list of where each command is covered. There is also a detachable fold-out UNIX reference card, which summarizes common UNIX, Bourne shell and *vi* commands.

Both books accomplish their purpose admirably. Taylor works in somewhat greater depth and includes an introduction to X and the use of the VP/ix utility from Interactive Systems to run DOS as a task under UNIX. Topham makes use of diagrams and computer output, as well as the command summary and reference card. Neither of these books would be a mistake to buy but I recommend Topham's book for DOS users who need to go deeper into UNIX's organization and subsystems.

***Stephan M. Chan** is president of Uniprime Systems, Inc., a computer consulting firm in Baltimore, MD, which specializes in systems design engineering involving UNIX.*

Technical UNIX® User Group

TUUG Business Meeting Minutes

Tuesday, January 8, 1991

Attendance:

16 present including a number of potential new members.

Minutes of January Meeting:

Reviewed and accepted.

Membership Secretary's Report:

No new members since the last meeting. Dave Long requested a membership list, perhaps to be included in the last newsletter before the summer break.

Newsletter Editor's Remarks:

Susan Zuk was absent, but new material is always welcome. Susan has announced that she will be stepping down as newsletter editor, probably after the June issue. (Nominations, suggestions or volunteers for the position are therefore welcome. Susan has been doing an excellent job as editor and will be missed.)

Treasurer's Remarks:

No change in financial status.

Meeting Coordinator's Remarks:

A presentation by HP is tentatively planned for April and a presentation on PC/NFS is tentatively planned for March. Suggestions and volunteers for presentations are always welcome.

Technical UNIX® User Group

Agenda

for
Tuesday, March 12, 1991
7:30pm
XCan Grain Ltd.
Toronto Dominion Building
1200-201 Portage Avenue
(See Page 2 - Announcements for Details)

1. Round Table 7:30

2. Business Meeting 8:00
    a) Minutes of February's Meeting
    b) Membership Secretary's Report
    c) Newsletter Report
    d) Treasurer's Report

4. Break 8:30

5. Presented Topic 8:40
    Open Desktop
    Presented By Eric Carsted

6. Adjourn 9:30