

*Technical UNIX® User Group*

October 1989  
Volume 2, Number 1

\$2.50

newsletter of the  
**Technical UNIX®**  
**User Group**

**This month ...**

The President's Corner

Book Review - UNIX Administration Guide

Solving Those Puzzling Quirks of UNIX Systems Use

Membership Renewals

Minutes from Sept 12 Meeting

Agenda for Oct 10 Meeting

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

# ***Your article could go here!***

**Become an active member of the Technical  
UNIX User group and submit your  
newsletter article today.**

---

## **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:	Vacant	
Treasurer:	Gilles Detillieux	261-9146
Secretary:	Susan Zuk	(W) 786-8483
Newsletter Editor:	Darren Besler	(W) 941-2976
Membership Sec.:	Pat Macdonald	(W) 474-9870
Information:	Gilbert Detillieux	261-9146
	(or) Susan Zuk	(W) 786-8483

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 Change:**

The September meeting location will be provided by UNISYS Canada Inc., Suite 300-1661 Portage Ave (UNISYS Building). Upon entering the building you will then be required to sign-in. Please sign-in using "TUUG" as the agency represented.

# President's Corner

by Gilbert Detillieux, President

Well the big news this month is the elections for the new executive for the group. Unfortunately, it looks like none of the positions will be contested, not even the treasurer (we'll miss you Richard). So the slate of candidates, as it now stands, is as follows:

President: Me  
Vice-president: Derek Hay  
Secretary: Matt Binnie  
Treasurer: Gilles Detillieux  
Membership sec.: Pat Macdonald  
Newsletter editor: Susan Zuk  
Meeting co-ordinator: Matt Binnie  
and the executive

All of these candidates are likely to win by acclamation, at the October "elections." Hope you'll all show up at that meeting to show your support for the new executive.

The October meeting is also the deadline to renew your membership. This edition of the newsletter is the last you will receive, unless you renew by then. The fee remains \$20 for the whole year, and goes mainly toward postage and stationery cost for the newsletter. If you feel the meetings and the newsletter are worthwhile, I'm sure the fee is very reasonable,

so don't delay - renew today! If you can't make it to the next meeting, just fill out the handy application/renewal form that should be enclosed with the newsletter, and mail it in along with your cheque.

Incidentally, if any of you have already paid for your renewal, but haven't filled out one of the application forms, please do so. We want to update our database with this information, for all our current members, not just the new ones.

Last month, I mentioned some of the ideas the executive was working on for the new year, such as setting up a UUCP network, starting an Industry SIG, and getting more involved with the MWUG (it's BBS has a UNIX conference and some public domain UNIX software, which we should check out). Once the elections are over, the new executive will meet to plan out the next year. We will keep you posted of any plans as they develop.

The October meeting will be held at Unisys again. The presented topic, after the business meeting, will be the ever-popular System Administration Workshop. Several ideas were brought up by members at the last meeting, and we will likely follow up on some of those at this meeting. Hope to see you all there!

---

## The fortune file

This month's fortune comes courtesy of Gilbert Detillieux who found it on a DOS program called MURPHY.

```
''You can lead a man to slaughter, but you can't make him think.''
```

---

## Quick Tip

Submission from Kirk Marat:

```
#  
# Isuser descends a directory and finds "where is file of user in dir."  
#  
# AUTHOR M. Engelhardt .. /23/2/89  
# Bruker Analytische Messtechnik GmbH, West Germany  
case $# in  
1) find . -depth \(-name '*' -user $1 -type f -exec ls -og {} \; \;);  
3) find $3 -depth \(-name '*' -user $1 -type f -exec ls -og {} \; \;);  
*) echo wrong usage: 'Isuser user' or 'Isuser user in dir2'  
esac
```

# UNIX Administration Guide for System V

by : Rebecca Thomas and Rik Farrow  
Prentice Hall, Englewood Cliffs, New Jersey  
ISBN 0-13-942889-5  
636 pages  
\$49.99 CDN (summer 1989)

*Reviewed By Kirk Marat, Dept. of Chemistry,  
University of Manitoba*

At one time complex operating systems such as UNIX ran only on large "mainframes" or mini-computers cloistered away in computer centers. The set-up and day to day administration of these systems was left in the hands of system administrators and the average user would simply login and use whatever utilities he or she needed. With the current boom in mid-range workstation computers, a UNIX box is as likely (or probably more likely) to be on an engineer's desk or in a chemist's lab as in a computer center. In many such cases the user must also now administer the system. Even when used by a single person the UNIX operating system still requires a fair bit of care and feeding. This book is ideal for the user of a small UNIX based computer who wants to spend as little time as possible staring blankly at the system manuals. After all, computers were supposed to make us more efficient, right?

As the title implies, this book is intended primarily for users of AT&T system V. BSD users will still find the book useful, however, and the authors point out critical differences between the AT&T, Berkley and other UNIX versions in a number of places. The book is not an introduction to the UNIX system and pre-supposes a knowledge of the basic UNIX commands. You do not, however, have to be a computer professional to benefit from it. Indeed, the people most likely to need this book are the owners/users of small UNIX systems serving 1 to perhaps a dozen users, and who do not have a professional UNIX administrator. If you are setting up a UNIX based workstation it would be a wise move to read this book even before plugging the computer in. These machines are not simply fast PCs.

Chapter 1 gives a brief history of UNIX and describes the essential configuration files and daemon processes. Also included are short tours of a typical file system and the AT&T documentation. Chapter 2 is called "Account Management" and describes in great detail how to add a user to the system or remove a user from the system. It does not cover system accounting (such as billing users), which is left to Chapter 7. Chapter 3 describes file system backup procedures while

Chapter 4 covers creating and maintaining file systems. System startup and shutdown are covered in Chapter 5. It is my opinion that this material should have been included earlier, possibly as Chapter 2. Chapter 6 covers the LP system including hardware connections and sample interface scripts. UNIX system accounting is covered in Chapter 7, but is probably not that useful to users of single user workstations who often turn the accounting system off. Chapter 8 is probably the best description of setting up UUCP communications that I have seen. It is a thorough (over 100 pages) step-by-step guide to setting up UUCP, covering both the older "Version 2" UUCP as well as the newer HoneyDanBer (HDB) UUCP. This chapter includes a very good description of RS232 serial interfacing to UNIX and describes how UNIX uses the various handshake and modem control lines. The RS232 section is worth reading even if you are not setting up UUCP but merely connecting login terminals and/or modems. Chapter 9 is devoted to system security and gives suggested access permissions for all sensitive system files. Also included are sections on encrypting files, understanding the system breakers tactics and defeating the system breaker. "Watchdog" scripts for ferreting out possible security breaches are also included. Common system problems such as terminal lockup, runaway programs and mysterious error messages (such as: "Panic: no imt", or how about: "proc on q") are the subject of the final chapter. This chapter also shows you how to set up an alternate root file system (with the appropriate security warnings!) and how to replace a lost root password. Nine appendices cover such odds and ends as environment variables, utilities for account management, predetermined C shell variables, and a restricted environment for guests.

"UNIX Administration Guide for System V" is an excellent book for anyone involved in any fashion with UNIX administration. It is well written, lucid and quite readable (quite unlike most UNIX documentation). Most chapters contain a handy reference summary at the end, and the book contains numerous handy (Bourne) shell scrips and C programs. It can put the installation Guides and Administration Guides of some (many?) vendors to shame.

# Solving Those Puzzling Quirks of UNIX Systems Use

By Steven List and Bruce Syewart

Reprinted from /usr/group CommUNIXations January/February 1988

One of the more powerful components of the UNIX system is the full-screen editor, *vi*. Unlike many commercial products, *vi* was never intended to be a full-featured wordprocessor. Instead, it was designed to suit the specialized needs of programmers. Unfortunately, much of the power and flexibility of *vi* is hidden behind its cryptic user interface and terse documentation. In this month's column, we offer a few examples of the tricks and hints we've acquired in using the *vi* editor. We've concentrated on one command - the exclamation point - as a sample of the power available from within *vi*.

## More Bang for the Buck

One of the more useful commands in *vi* is the exclamation mark (bang). It occurs in several forms: the colon-bang (:!), the embedded-bang (...!...), the bangbang (!! ) and the bang-move (!<move>). Although each form of the command accomplishes a different purpose, the bang in each performs the same task - a shell escape. The specified commands are executed in another shell, separate from the one in which the command was invoked.

Most users of *vi* are familiar with the basic form of the command, the colon-bang or shell escape. With this command, any shell command may be executed from within *vi*. Of course, the drawback is that the results of the command are usually not available when the command finishes. Results must be saved into a file for later retrieval. For example, you might want to include a list of the files in the current directory in the file you are editing. With the colon-bang, you could do it in this way:

```
:w file
:!ls -l >> file
:e! file
```

Of course, with this approach the listing is appended to the end of the file and you still must move it to where you want it. An alternative is to save the output of the *ls* command to a file and then read it in where you want it. You might try:

```
:!ls -l > jnk
:r jnk
```

This will read the long listing directly into the file after the current line. Although better, this method still requires too many steps and creates an unnecessary disk file. Instead, you could use the embedded-bang to produce the same result with a single command:

```
:r !ls -l
```

This command first executes the shell command, *ls -l*, and then reads the output of the command into the file after the current line. Besides being only a single command, this method produces no extra disk file. Although this is a simple example, it demonstrates the power of the embedded-bang.

The next bang command is the bang-bang or double-bang. With this command, the output of a shell command replaces the current line in the file. This command is useful for marking places in a file where you expect to add information later. The added information will replace the markers in the text. We will discuss two examples of using the double-bang command to replace markers in a file: the manual method and the automatic (macro-assisted) method.

If we were writing a column about the *ls* command, we might include samples of the output from each of its options. We might also compare those with the output of the command with no options. For each output example, we could insert a marker in the text, such as these:

```
###listnoopt
###listlong
```

Each marker identifies a location where output from the *ls* command will be imbedded in the text. You could use the manual method to replace these markers with examples of the output from the *ls* command with each option:

```
/###/
!!cat listnoopt
n!!cat listlong
```

These three lines say, in English:

*Find the next line containing three pound signs.  
Replace the current line with the contents of file "listnoopt."  
Find the next line containing three pound signs.  
Replace the current line with the contents of file "listlong."*

Although the manual method is satisfactory for a single occasion, it has several drawbacks. It requires a number of keystrokes to replace each marker with the appropriate output, and also assumes you already have available the files containing the output examples. You can improve this solution significantly by using the macro facility of *vi* to store your keystrokes

for repeated execution. We use the following macro:

```
:map ^A /###/^M:s/*###/!cat/^M"add`a
```

If you include this macro in your .exrc or other file, leave off the initial colon. For those who are not familiar with using the control characters in commands or files, those pairs above that have the form "^X" are inserted with the sequence : ctrl-V ctrl-X. Our macro maps the keystrokes ctrl-A to the following:

Find the string ###

Replace everything up to and including the ### with the string "!!cat"

Delete the current line into buffer "a" as a macro.

Thus, the line

```
###listnoopt
```

becomes

```
!!cat listnoopt
```

which is deleted into buffer "a" and executed. This causes the line with the original marker to be deleted. The line following the contents of the file "listnoopt." Because you have mapped the ctrl-A character to this macro, you can perform this double-bang command for each marked line with a single keystroke.

Of course, you can also improve on this solution by carefully choosing your labels and using the double-bang to execute the *ls* command. This technique will replace each marker with the output of the *ls* command. For example, you might select your markers so that each marker uses the option whose output you wish to capture. You could use markers such as those below.

```
###C      (the C option)
###a      (the a option)
###l      (the l option)
```

These identify the places in the text where command output would appear. With these changes, your macro becomes:

```
:map ^A /###/^M:s/*###/!!ls -/^M"add@a
```

This uses the double-bang to execute the *ls* command and embed the output in your text. You no longer need a separate, resident file containing a sample of the output for each option. This last solution also has its drawbacks, however. You cannot use the "yank" capability of *vi* within a macro, and so resort to deleting lines into buffer "a" instead. This causes you to lose a

line at each invocation of the macro. To compensate, you must follow your markers with an extra blank line for spacing.

Finally, consider the bang-move. This command allows you to do such things as sort the contents of a list within *vi*. Without the bang-move command, you would need commands such as the following to sort the next 12 lines of a file:

```
.,+11w sortin
:!sortsortin-o sortin
.,+11d
:r sortin
```

You can shorten these somewhat by using the double-bang to replace the current line with the output of the sort command. The bang=move, however, allows you to reduce the effort to a single command:

```
!11+sort
```

This translates to:

*Execute the shell command "sort" against this line and the next 11 lines. Replace the specified lines with the output of the shell command.*

If you read "!11+" as "bang-move down 11 lines," you will see the bang-move. The command "11+" can be any movement command. Thus, you can use marks (mx) to identify the boundaries of the text to be affected, and the mark movement commands (x or `x) to perform the move. You can use character, word, or line movement commands, as in our example. You can also use page or screen movement commands. You can use virtually any movement command for the move component of bang-move. This feature is the one that gives the bang-move its power because you can specify any group of lines to be operated on by the shell command.

As these examples illustrate, the bang commands are powerful and versatile features of *vi*. Strictly speaking, the bang commands aren't *vi* commands at all, but shell commands whose output may be captured by *vi*. Nonetheless, the ability to suspend the current *vi* session and execute a shell command can be a timesaver, and a convenient, effective way to exploit the power of both the shell and *vi*.

**Steven List is an independent consultant specializing in UNIX and shell programming, and in database management applications. Bruce Stewart is an independent software consultant experienced with programming, system administration and applications development.**



## Minutes From the Business Meeting September 12, 1989

### 1. Minutes:

MOTION : (Susan Zuk) The minutes from the May 9th, 1989 meeting be approved.

SECONDED : (Peter Somers)

In Favour : 11

Opposed : 0

Carried

### 2. Membership Report:

Membership renewals are due in October. Please fill out the membership forms you received with the newsletter. The October newsletter is the final newsletter of this year.

### 3. Newsletter Report:

Please contribute articles for the newsletter.

### 4. Treasurers Report:

The financial report will be prepared for the October meeting. At the present time approximate information is as follows:

Assets \$350.00

Liabilities \$200.00

Equity \$150.00

### 5. 1989-1990 Executive Nomiations:

President : Gilbert Detillieux

Vice President: Derek Hay

Treasurer:Richard Willacy

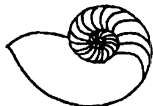
Treasurer: Gilles Detillieux

Newsletter Editor: Susan Zuk

Newsletter Editor: Darren Besler

Secretary:

Meeting Co-ordinator:



Technical UNIX® User Group

**Agenda**  
**for**  
**Tuesday, October 10, 1989**  
**7:30pm**  
**UNISYS**  
**UNISYS Building**  
**300-1661 Portage Avenue**

- |                                   |      |
|-----------------------------------|------|
| 1. Round Table                    | 7:30 |
| 2. Business Meeting               | 8:00 |
| a) Minutes of September's Meeting |      |
| b) Membership Secretary's Report  |      |
| c) Newsletter Report              |      |
| d) Treasurer's Report             |      |
| 3. Elections                      |      |
| 4. Break                          | 8:30 |
| 5. Presented Topic                | 8:40 |
| System Administration Workshop    |      |
| 6. Adjourn                        | 9:30 |