

*Technical  
UNIX®  
Buyers  
Association*

April 1990  
Volume 2, Number 7

\$2.50

Mr. Gilles Detillieux  
Consultant  
Info West Inc.  
P.O. Box 130  
St. Boniface MB R2H 3B4

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

**This month ...**

The President's Corner  
The Fortune File  
Programming Tips  
Smileys  
Employment Opportunity  
April Agenda

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

# Thoughts From The Chief PageMakerist

By Susan Zuk

This is our April Fool's Edition of the TUUG Newsletter. Enjoy the newsletter and make sure to laugh every so often.

This month we have a real grabbag of information. There are articles dealing anywhere from daemons to Programming Tips. Included are special National Enquirer type news clippings discussing interesting developments in the computer world.

For those of you who seem to be having problems with strange and crazy occurrences on your machine, we have some help for you located in the Fortune File.

To keep you "Smiling" there is an article called Smileys, which should keep you chuckling. In case you've never come across Smileys before, turn the page sideways while reading.

Our featured article is on the topic of the C shell. For those of you using the Bourne Shell environment, try the C Shell and see what you think of the different functionality. Try the history command, it is one of my favorite features of the C Shell. This command allows you to request previously typed commands. For those of us who don't have the best typing skills this really does come in handy. How is your typing when it comes to those long path names?

Something else which we have inserted into the newsletter is an employment opportunity from Ubitrex Corporation. Take a look to see if you are interested.

Well, that's all for now. Enjoy reading this April Fool's Edition and we'll see you at the next meeting. How about some Fortunes? Seen any lately?

---

## 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

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

# El Presidente Coronado

*By Gilberto Detillio*

UNIX is certainly hot news lately. New journals and publications focusing on UNIX are constantly springing up. Even the popular press is reporting on UNIX occasionally. With all this new information, it is hard to stay on top of everything that is happening in the UNIX community. As president of the largest UNIX user group in Winnipeg, I am constantly being bombarded with news items and information about UNIX from around the world. As a service to our members, and for your reading enjoyment, I now present some of the more obscure news items to cross my desk in the last few weeks. (The opinions expressed in these news items are not my own, nor do they represent those of TUUG as a whole.)

**Window Wars:** The competition in the area of windowing systems is about to get a lot hotter. MetaSoft Inc. (New Aura, CA) has announced "Pizzazz," a three-layered graphical environment, which it claims overcomes the limitations of all the previous systems. The base layer, a Universal Windowing Environment (UWE), simultaneously supports X11, NeWS, and Display PostScript. The middle layer, a Graphical User Interface (GUI), supports OpenLook, Motif, and NeXTStep, and allows users to select a window manager emulating their favorite. For application development, a third layer provides Graphical Object Oriented Development (GOOD) facilities. The combined product (UWE/GUI/GOOD) is touted as a windowing system "with the works!" However, market introduction may be hampered by extensive litigation; lawsuits have already been launched by Apple, Adobe, Microsoft, NeXT, Sun, and Pizza Hut. (Source: UNIX Tomorrow)

**Caveat Lector:** PC Magazine columnist and long-time UNIX critic, John C. Dvorak, has recently written a buyer's guide to UNIX systems, entitled "UNIX, My A\*\*!" (To limit the solution space, the wildcard expression could be written as "A?!!", or "A?N!" for C-shell users.) Demand for the book has so far been rather low, and mostly limited to other PC Magazine columnists. (Source: UNIX After Dark)

**Digital Sells Its Employees:** Digital Equipment Corporation announced today that since it has had so much trouble selling its VAXes and has so many excess employees, it will begin offering the employees for sale. According to our reporter, DEC expects its employees to run about 20%

faster than its older machines and about half the speed of its RISC-based DECStation line. Prices and delivery are expected at the NODECEMP show in January. (Source: DEC Confuser News)

**Standards 'R' Us:** Sun Microsystems recently pronounced that X-Windows is not an industry standard. A Sun spokeswoman claimed that it couldn't possibly be a standard as Sun has not invented it yet. "As soon as we come up with it, it will be made a standard" she said. "We expect to have a working X-Windows before the turn of the century and we promise to be first to market as we always are." (Source: DEC Confuser News)

**DEC Introduces Its First Mainframe:** Digital Equipment Corporation announced its first mainframe size computer. "We tried to make it bigger than IBM's but we did not have enough sheet metal in the plant" a DEC spokesman said. "However, it does out run the DECStation 3100 by 20% and is at least twice as fast as the employees we will be selling in the next year." (Source: DEC Confuser News)

\* \* \* \* \*

Now on to more serious stuff. For those of you who missed it, the March meeting featured a look at VP/ix on a PC/AT clone running Unix 386/ix, presented by our vice-president, Derek Hay. For the April 10 meeting, the topic is yet to be determined, but will either be a look at Oracle, or more about networking. We will be meeting at the new Unisys offices (10th floor, same building: 1661 Portage, between St. James St. and Route 90), starting at 7:30 PM sharp.

The May meeting will likely feature a discussion on Graphical User Interfaces, and a look at some graphics workstations. The location for this meeting will likely be the U of M Computer Science department, but has yet to be confirmed, so stay tuned. As for the June meeting, which will be the last one before the summer (we meet again in September), we will likely continue with the "tradition" started last year, and have a barbeque. The location and all other details have yet to be arranged. Any ideas or offers? Please let us know.

Hope to see you at the next meeting (April 10). A happy springtime and a happy Easter to all!

# UNIX Programming Tips

By Curt Motola & Bruce Stewart

Reprinted with permission from the March/April issue of *CommUNIXations*,  
published by Uniforum

Most UNIX users prefer the C shell (csh) as their login shell. The C shell offers a more complete set of interactive features than the Bourne shell, the standard shell supplied with all UNIX systems. C shell users can take advantage of features such as command aliasing, job control, file name completion, and command history and simple command line editing. And although the Korn shell has a comparable set of interactive features, it is not as widely available as the C shell.

The C shell environment is the product of the efforts of two people: the UNIX system administrator and the individual user. The system administrator establishes the global C shell environment by creating or modifying a file that sets system-wide C shell parameters. Users may customize the environment by placing C shell commands in the `.login` and `.cshrc` files in their home directory. During login, the C shell reads and processes commands in each of these files.

After a look at how the C shell environment is built, we'll suggest how individual users can modify their environment. Since some of the steps required to set up the C shell environment are implementation dependent, a general discussion should apply to most UNIX installations.

Construction of the C shell environment begins during login. The login program reads the `/etc/passwd` file and sets the environment variables `HOME`, `LOGNAME` (sometimes `USER` instead of `LOGNAME`) and `SHELL` from information contained in the password file. The login program also sets the `PATH` variable and may set the `MAIL` variable. The default value for the `PATH` variable is implementation dependent but typically includes `/bin` and `/usr/bin`. An easy way to determine the default values for the variables set by your login program is to add a new user to the password file whose login shell is `/bin/env` or `/bin/printenv`. When this user logs in, the shell variables set by login are printed and the user is logged off.

After setting these environment variables, the login program invokes the program specified in the `/etc/passwd` file or the Bourne shell if no program has been specified. Note that this program could be an application or a UNIX

Figure 1: Global cshrc file

```
# set variables that don't get inherited by subshells
# That is, for login shell only

# log user out after 30 minutes of inactivity
set autologout=30

# set command history to 20
set history=20

# set up a numbered prompt
set prompt="[N]"

#
# set variables that will be inherited by all subprocesses
#
# make -rw-r--r-- the default file creation permissions
umask 022

# force "logout" command to be used to log out
# that is, don't accept ^D
set ignoreeof

# append to PATH directories with commands for
# all users
setenv PATH "$PATH":/usr/local/bin

# for system 5-based systems - set timezone
# (e.g., Pacific Daylight Time)
setenv TZ PST8PDT

# Some versions of remote login (rlogin) propagate
# TERM to the new login. Therefore, set TERM only
# when it is not set. In this case, set it to the
# default depending on the port being used.
if ( ! $?TERM ) then
    set noglob
    eval `tset -s -h`
    unset noglob
endif

# if "USER" not defined, set it to "LOGNAME" since
# shell scripts often refer to "USER"
if ( ! $?USER ) setenv USER "$LOGNAME"

#
# Display information to the user
#
# display the message of the day
if ( -r /etc/motd ) cat /etc/motd

# tell user if mail is waiting
if ( -x /bin/mail ) then
    /bin/mail -e
    if ( $status == 0 ) echo "You have mail."
endif
```

utility, and does not have to be a standard interactive shell. By convention, the program invoked during the login process is called the login shell. In all cases, the C shell is invoked (by login) in a special manner that tells the shell that it is a login shell. When this program (csh, for our purposes) begins execution, it recognizes that it is a login shell and performs special processing to establish the environment.

As a login shell, the C shell first reads the file that establishes environment variables and does processing desired for every user under the C shell. Examples of this processing are displaying the message of the day, and notifying the user of recent news and, if appropriate, of new mail.

The name of this file varies with the version of UNIX; frequently the file is named /etc/csh.login, /etc/cshrc, or /etc/cshinit. The C shell reads this file only when invoked as a login shell. This file contains commands that specify the minimum environment for all users and typically sets default values for environment variables such as TERM and PATH. Figure 1 shows an example of this type of file.

After processing the global cshrc file, the C shell changes directory to the user's HOME directory and processes the .cshrc file if one is present. This file is processed whenever a new C shell is created, whether or not it is a login shell. The .cshrc file usually contains commands to tailor the environment to the preferences of the individual user, typically, these are commands that should be executed whenever a new C shell is spawned. Command aliases, length of the history list and prompt string are commonly set in the .cshrc file.

Since this file is processed for each new C shell spawned, commands in the file that invoke other programs or scripts can adversely affect the performance of C shell scripts. In general, this file should contain commands that do not result in lots of processing. Figure 2 shows a sample .cshrc file.

Finally, the login C shell reads and processes the .login file in the user's HOME directory. This file is read only during login, and typically contains commands to set environment variables and terminal characteristics that only need to be set once. Most environment variables are in this category. Figure 3 shows a sample .login file.

When the .login file has been processed, login is complete and the user sees the C shell prompt. The C shell environment is set but not fixed. Users are free to modify their

**Figure 2: .cshrc file**

```
# save last 200 commands available via history command
set history=200

# keep record of last 200 commands in .history
set savehist=200

#useful aliases (command rename feature)
alias h 'history -r | more'
alias l 'ls -aCF'
alias ll 'ls -alF'
alias tl 'tail -f'
alias m more
alias j jobs
alias pd pushd
alias p pwd

# prompt contains a numbered prompt and the name of
# current directory (not the path of the current directory)
# not all C shells set "cwd" variable when "cd" is performed
alias cd 'cd \*; set prompt="$cwd:t_[\N] "'

# cd is required to activate the directory information in
# the variable prompt, for the first prompt
cd .
```

**Figure 3: .login file**

```
# set ERASE sequence to clear character being erased
# for System 5-based systems
stty echoe
# for BSD-based systems
stty crt

# add personal command directories to global command
# directories
setenv PATH "::$HOME/bin:$PATH"

# every 60 seconds notify user of new mail
set mail=(60 /usr/spool/mail/$USER)
# other systems, use
set mail=(60 /usr/mail/$USER)
```

environment with the appropriate C shell commands (for example, setenv and unsetenv).

These commands are particularly useful when running C or Bourne shell scripts that depend upon the values of environment variables to control their operation. Fortunately, the C shell manages some environment variables automatically to provide a consistent environment for executing either Bourne or C shell scripts.

When a C shell is invoked, the variables USER, TERM, PATH and HOME are checked. If the variable is defined, the value is mapped to the C shell variables user, term,

path and home, respectively. If any of these C shell variables are changed, its new values are immediately copied to the corresponding USER, TERM, PATH or HOME variable. Thus, a Bourne shell script invoked from the C shell will be executed with the latest values of the USER, TERM, PATH and HOME variables. Further, when a C shell script is invoked from a Bourne shell, the C shell variables users, term, path, and home get set to the values of their Bourne shell counterparts.

However, establishing the environment for execution of a shell script is only the first step. The proper command interpreter (shell) must still be called to process the commands in the script. The script itself is examined in order to determine the correct shell to invoke. If the first two characters of the script are #!, the remainder of the first line is taken as the command (possibly with options) to interpret the script.

For example, #!/bin/csh or #!/bin/csh -f will each cause the C shell to be called to interpret the script. If the first character is # and the second character is not !, then the shell specified in the variable called shell is used. If this variable is not specified, then /bin/csh is used. Finally, if the first character is anything other than #, the Bourne shell is called.

When the C shell is invoked to process a shell script, it first reads the .cshrc file in the user's HOME directory to establish the environment for executing the script. Thus, commands in the .cshrc file may reset the values of environment variables changed interactively and produce unexpected results when the script is executed. The -f ("fast") flag can be used to invoke a C shell without reading the .cshrc file. Also note that aliases defined in the .login file are lost, since this file is read only for a login C shell.

The above constitutes a fairly generic process for establishing the UNIX C shell environment. Because of the variety of UNIX implementations, we cannot hope to cover every variation. Nevertheless, these general criteria should help in determining what commands to place in the various files used by the C shell to establish the environment.

**Curt Motola is president of Classic Software, Inc., a supplier of UNIX consulting services. Prior to founding CSI, he held several engineering and management positions with Hewlett-Packard Company and Benetics Corporation. Bruce Stewart is an independent computer consultant experienced in program development, systems administration and system design.**

## The Fortune File

Submitted by The UNIX Exorcists

*Plagued by Daemons?*

*Is your UNIX system acting erratically? Is its performance being sucked away by mysterious forces? We can help free your system from those nasty daemons! Call today for a free computer aura check.*

*The UNIX Exorcists  
1-800-DAE-MONS*

## Employment Opportunity

### Software Designer

Ubitrex Corporation is developing a world class product targeted at the needs of the nursing community. We are creating products employing the latest workstation, networking, software and microelectronics technology. By joining our team, you can help create a new generation of products for the North American health care marketplace. We require a software designer to assist in the development of a product which requires a unique blend of high speed transaction-oriented database design, networking and an innovative user interface. You will have the opportunity to work with the product from its early stages through to market introduction.

Applications are invited from individuals with a degree in Computer Science and experience in two or more of the following technologies:

- Unix (on Sun workstations is an asset)
- C language development
- Object oriented programming (with C++ an asset)
- Graphic user interface development (with SunView or X-Windows an asset)
- High throughput RDMS (SQL an asset)
- Clinical systems development
- Software testing and quality assurance

This is an early stage opportunity with a company on the leading edge of a major new market in the health care field. We offer a complete range of benefits including relocation assistance and a stock option plan. Resumes can be submitted in complete confidence to Ubitrex Corporation, 221-435 Ellice Avenue, Winnipeg, Manitoba, R3B 1Y6.

# Smileys

## A Glossary

*Compiled by Gilles Detillieux*

There are several lists of "smileys" - those little faces people put in their e-mail messages - making their rounds on the Usenetnews feeds. Since we at T.U.U.G. take pride in bringing you news of important developments in the UNIX community, we now bring you this glossary, to help you better understand the e-mail you receive.

```
:~o    Wow!
:~|    Grim
:= |   Baboon
:~v    Speaking
:~V    Shout
:~w    Speak with forked tongue
:~r    Sticking tongue out
:~*    Oops!
:~T    Keeping a straight face
:~D    Said with a smile
:~x    Kiss kiss
:~[    Pouting
:~X    A big wet kiss!
:~Y    A quiet aside
>~<   Absolutely livid!!
:~c    Real unhappy
:~C    Just totally unbelieving
:~B    Drooling
:~,    Smirk
:~||   Anger
:~)    Smiling
:~(    Frowning
`~)    Wink
;~)    Sardonic Incredulity
%~<I>  Drunk with laughter
:~''   Pursing lips
:~#    My lips are sealed
:~P    Tongue hanging out in anticipation
8~|    Eyes wide with surprise
&~|    Tearful
|~{    ``Good Grief!'' (Charlie Brown?)
:~}    ``Thish wine tashted pretty good''
8~]    ``wow, maaan''
8~O    ``Omigod!!'' (done after ``rm -rf *'' ?)
:~,    ``Hmmm.''
|~(    ``Someone just busted my nose''.
:~^D   ``Great! I like it!''
B~D    ``Serves you right, dummy!''
```

Here's another list, "The Unofficial Smiley Dictionary", from IBM SHARE/Northeast Regional Data Center, Gainesville, Florida. (Updated May 1985.)

:~) Your basic smiley. This smiley is used to

inflect a sarcastic or joking statement since we can't hear voice inflection over Unix.

```
;~)    Winky smiley. User just made a flirtatious and/or sarcastic remark. More of a ``don't hit me for what I just said'' smiley.
:~(    Frowning smiley. User did not like that last statement or is upset or depressed about something.
:~I    Indifferent smiley. Better than a Frowning smiley but not quite as good as a happy smiley
:~>    User just made a really biting sarcastic remark. Worse than a ;~).
>~>    User just made a really devilish remark.
>~>    Winky and devil combined. A very lewd remark was just made.
```

Those are the basic ones... Here are some somewhat less common ones:

```
(~:    User is left handed
%~)    User has been staring at a green screen for 15 hours straight
:~*    User is drunk
[:]    User is a robot
8~)    User is wearing sunglasses
B:~)   Sunglasses on head
:~:)   User wears normal glasses
B~)    User wears horn-rimmed glasses
8:~)   User is a little girl
:~)~8  User is a Big girl
:~{)   User has a mustache
:~{}   User wears lipstick
{:~)   User wears a toupee
}:~(   Toupee in an updraft
:~[    User is a Vampire
:~E    Bucktoothed vampire
:~F    Bucktoothed vampire with one tooth missing
:~7    User just made a wry statement
:~*    User just ate something sour
;~(    User is crying
;~)    User is so happy, s/he is crying
:~@    User is screaming
:~#    User wears braces
:~^    User has a broken nose
:<~)   User is from an Ivy League School
:~&    User is tongue tied.
+~:~)  User is the Pope or holds some other re-
```

```

ligious office
`:-) User shaved one of his eyebrows off this
      morning
,-:-) Same thing...other side
|-I   User is asleep
|-O   User is yawning/snoring
:-Q   User is a smoker
:-?   User smokes a pipe
O-)   Megaton Man On Patrol! (or else, user is
      a scuba diver)
O :-)  User is an angel (at heart, at least)
:-P   Nyahhhh!
:-S   User just made an incoherent statement
:-D   User is laughing (at you!)
:-X   User's lips are sealed
:-C   User is really bummed
<|-)  User is Chinese
<|-(  User is Chinese and doesn't like these
      kind of jokes
:-/   User is skeptical
C=-:-) User is a chef
@=    User is pro-nuclear war
*<:-) User is wearing a Santa Claus Hat
:-o   Uh oh!
(8-o  It's Mr. Bill!
*:o)  `And Bozo the Clown!
3:]   Pet smiley
3:[   Mean Pet smiley
d8=   Your pet beaver is wearing goggles and a
      hard hat.
E--:-) User is a Ham radio operator
:-9   User is licking his/her lips
%-6   User is braindead
[:-)  User is wearing a walkman
(:I   User is an egghead
<:-I  User is a dunce
K:P   User is a little kid with a propeller
      beanie
@:-)  User is wearing a turban
:-Ø   No Yelling! (Quiet Lab)
--:   Mutant Smiley
      The invisible smiley
.-)   User only has one eye
,-)   User only has one eye...but he's winking
X-(   User just died
%\v   Picasso
C=}>)*{O) .. Mega-Smiley... A drunk, devilish
      chef with a toupee in an updraft, a
      mustache, and a double chin

```

Note: A lot of these can be typed without noses to make midget smileys.

```

:]    Gleep...a friendly midget smiley who
      will gladly be your friend
:)    Happy
:>    hmm, let me think...
:D    Laughter
:I    Hmmm, not funny!
:(    Sad
:[    Real Downer
:<    what pretences!

```

```

{:    oh boy, the headmaster!...
:O    Yelling
;(    Crying
[]    Hugs
:*    Kisses

```

Here are still more of them.

```

:-`   smiley spitting out its chewing tobacco
:-1   smiley bland face
:-!   smiley bland face
:-$   smiley face with it's mouth wired shut
:-6   smiley after eating something sour
8-)   smiley swimmer
:-*   smiley after eating something bitter
:-&   smiley which is tongue-tied
:-0   smiley orator
(:-(  unsmiley frowning
=:-)  smiley punk-rocker
=:-(  (real punk rockers don't smile)
+--:-) smiley priest
:-o   smiley singing national anthem
:-p   smiley sticking its tongue out (at you!)
:-[   un-smiley blockhead
:-]   smiley blockhead
:-{   smiley variation on a theme
:-}   smiley variation on a theme
{:}   smiley with its hair parted in the middle
}:-)  smiley with its hair parted in the middle
      in an updraft
g-)   smiley with pince-nez glasses
:-\   undecided smiley
:-/   lefty undecided smiley
:-|   ``have an ordinary day'' smiley
;-)   winking smiley
:-<   real sad smiley
:-x   ``my lips are sealed'' smiley
:-c   bummed out smiley
:-v   talking head smiley
:-?   smiley smoking a pipe
0-)   smiley cyclops (scuba diver?)
:<    midget unsmiley
:>    midget smiley

```

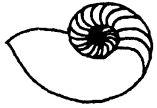
Here is yet another smiley list (YASL):

```

:-)   ha ha
|-)   hee hee
|-D   ho ho
:->   hey hey
:-(   boo hoo
:-I   hmm
:-O   uh oh
:-P   nyah nyah
|-P   yuk
~:-(  net.flame
X-(   net.suicide
>:-I  net.startrek
3:o[  net.pets
:-    male
>-    female
8     infinity

```





Technical UNIX® User Group

**Agenda**  
**for**  
**Tuesday, April 10th, 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 |
| DOS/UNIX Interfacing             |      |
| OR                               |      |
| System Administration            |      |
| 6. Adjourn                       | 9:30 |