



Technical UNIX®Users Group

March 1989
Volume 1, Number 6

\$2.50

newsletter of the
Technical UNIX®
User Group

This month ...

The President's Corner
The Ten Commandments for C Programmers
Broadcast News, The Script
Checking Users' Disk Quotas
Agenda for March 12 Meeting

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

President's Corner

by Gilbert Detillieux, President

Another month, another newsletter, already? Time sure does seem to fly by lately, and it sure doesn't help when the month is as short as February was. If time passes any more quickly for me, I'll have to get the type of wristwatch that a friend of mine always wanted to see: one that has the months and days around the dial, and tells the time of year, rather than the time of day, such as "half past January," or "2 days to March."

With the new month pouncing on us while we weren't looking, we were once again rushed to put this newsletter together. Some day, we keep saying, we'll have everything ready in advance, and be able to put together the newsletter without any panic. (This would have gone well with my other New Year's resolutions, if only I had not put those together in such a hurry!) Despite the last minute panic, we have a few articles for you this month, including a couple of shell scripts that may be useful to you. One is an implementation of the "news" command, that was discussed at the last meeting. The other is a little routine that checks for users who may be exceeding their quota of disk space. There are also some other articles "in the pipeline" right now, and hopefully, a couple will be ready for this issue.



At the last meeting, one member asked about telecommunication programs for UNIX systems, with file transfer protocols. For MS-DOS systems, there are a lot of packages to choose from, including several quite good shareware and public domain packages. For UNIX systems, there is always the old stand-by, "CU." But if you need something more sophisticated, there is a program called Professional-YAM (yet another modem?), available from Omen Technology. This is the company that developed the ZMODEM file transfer protocol, and the MS-DOS programs DSZ and ZCOMM, which implement this protocol. Here is the address, for those interested:

Chuck Forsberg
Omen Technology Inc.
17505-V Northwest Sauvie Island Road
Portland, Oregon 97231

Modem: 503-621-3746 Speed 1200,2400,19200 (PEP)
Compuserve: 70007,2304
GENie: CAF
UUCP: ...!tektronix!reed!omen!caf

Our February meeting continued the system administration workshop from the previous month. The discussion was again lively, and ran past 10:00 PM. Since these workshops seem to go over so well, and since we don't have any other prepared topic for the next meeting, we'll be back at Unisys (300-1661 Portage, between St. James St. and Route 90) to continue the system administration workshop. We covered a lot about system tuning last month, so we could move on to other topics this month, such as:

1. System accounting (process, login accounting, etc.)
2. Communications
 - a) Serial terminals (stty, gettydefs, terminfo)
 - b) Printer setup (lp, lpadmin, custom interfaces)
 - c) UUCP (use, setup, HDB-UUCP)
 - d) Networks (Ethernet, token ring, X.25, Usenet)
 - e) Interconnecting PC's and UNIX systems
3. Security (passwords, file permissions, etc.)

As was done at the last meeting, we will make sure we have someone in the lobby to let you all in; but if you arrive late, you may be out of luck if the security guard isn't at the front desk.

Hope to see you at the next meeting (Mar. 14). A happy St-Patrick's day and a "hoppy" Easter to all!

The fortune file

This month's fortune comes care of Darren Besler.

Do you have lysdexia?

The Ten Commandments for C Programmers

by Henry Spencer

- 1 Thou shalt run lint frequently and study its pronouncements with care, for verily its perception and judgement oft exceed thine.
- 2 Thou shalt not follow the NULL pointer, for chaos and madness await thee at its end.
- 3 Thou shalt cast all function arguments to the expected type if they are not of that type already, even when thou art convinced that this is unnecessary, lest they take cruel vengeance upon thee when thou least expect it.
- 4 If thy header files fail to declare the return types of thy library functions, thou shalt declare them thyself with the most meticulous care, lest grievous harm befall thy program.
- 5 Thou shalt check the array bounds of all strings (indeed, all arrays), for surely where thou typest "foo" someone someday shall type "supercalifragilisticexpialidocious".
- 6 If a function be advertised to return an error code in the event of difficulties, thou shalt check for that code, yea, even though the checks triple the size of thy code and produce aches in thy typing fingers, for if thou thinkest "it cannot happen to me", the gods shall surely punish thee for thy arrogance.
- 7 Thou shalt study thy libraries and strive not to re-invent them without cause, that thy code may be short and readable and thy days pleasant and productive.
- 8 Thou shalt make thy program's purpose and structure clear to thy fellow man by using the One True Brace Style, even if thou likest it not, for thy creativity is better used in solving problems than in creating beautiful new impediments to understanding.
- 9 Thy external identifiers shall be unique in the first six characters, though this harsh discipline be irksome and the years of its necessity stretch before thee seemingly without end, lest thou tear thy hair out and go mad on that fateful day when thou desirest to make thy program run on an old system.
- 10 Thou shalt foreswear, renounce, and abjure the vile heresy which claimeth that "All the world's a VAX", and have no commerce with the benighted heathens who cling to this barbarous belief, that the days of thy program may be long even though the days of thy current machine be short.

Broadcast News, The Script.

*Written by Gilles Detillieux.
Narrated by Gilbert Detillieux.*

The last system administration workshop included a digression into the use of the UNIX news command, which allows users to read news items submitted by the system administrator. This command keeps track of the date a user last received news, and usually lists only items submitted since that date. To help illustrate how the news command works, and to provide a functional equivalent for systems that don't have this as a standard command, we now present the news shell script.

The shell script is written for use with the Bourne shell, and makes use of common UNIX commands such as ls, find, and sed. It should therefore easily run on even the most stripped-down UNIX system. It provides all the basic functions of the news command, including the -a, -n, and -s options.

The script starts (on line 12) by changing to the /usr/news directory, where news items will be stored, one per file. If the directory is not accessible, the script just exits. Lines 13 to 15 initialize variables that will be used later. (It is a good idea to explicitly set any variables needed in a script to a null string initially, in case that variable already appeared in the environment.) Lines 16 to 24 then scan the argument list for options. The -a option will force the listing of all news items, rather than just the new ones. The -n option lists only the names of the news items (files), and the -s option indicates only the number of items. If any invalid option is encountered, a usage message is printed (line 21). The script stops scanning for options as soon as a non-option argument is encountered (line 22).

If any arguments exist, they are assumed to be the names of news items to be listed. If there were no arguments, lines 27 to

36 prepares a list of item names. First, line 27 produces a reverse chronological list (newest first) of file names, excluding the name "core," if it exists. Next, if the -a option was not specified, the list of file names is reduced to include only those modified since the last time this user ran the script (lines 29 to 33). The modification time of the empty file .news_time in the user's home directory is used to determine the time of the last run, and the find command is used to find the newer files. Line 36 then sets this list as our argument list.

If there are no items to list, the script just exits (lines 40 to 42). If the -n option was specified, the list of items is simply echoed (line 45). If the -s option was specified, the number of items is determined (line 46), and printed (lines 47 to 50).

Otherwise, for each item to be listed, the script prints out a header of the following form:

```
item-name (user) date
```

where item-name is the file name of the news item, user is the owner of the file, and date is the modification date of the file (lines 61 and 62). The contents of the file are then printed, indented by 3 spaces (line 64).

Finally, if the script was called with no arguments or options, the .news_time file is touched (line 70) to indicate that all news items up to the current time have been received. (This will create an empty file, if one didn't exist yet, or just update the modification time of the existing file).

Listing 1: Shell script for news command.

```
1  #! /bin/sh
2  #
3  # news.sh - print news items
4  #
5  # usage:      news [-a] [-n] [-s] [items]
6  #           where: -a      print all news items, don't update time
7  #                 -n      print names of current items, don't update time
8  #                 -s      print number of current items, don't update time
9  #                 items   print specified items, don't update time
10 #           (no args)  print all current items, update time
11 #
12 cd /usr/news || exit
13 time=
14 all=n
15 mode=
16 while :
```

```

17 do
18     case "$1" in
19         -a)      all=y; shift ;;
20         -[ns])  mode="$1"; shift ;;
21         -*)     echo "Usage: news [-a] [-n] [-s] [items]" >&2; exit 1 ;;
22         *)     break ;;
23     esac
24 done
25
26 case "$#" in
27 0) items="\ls -t | sed '/^core$/d'"
28     case "$all" in
29     n) time=$HOME/.news_time
30         if [ -r $time -a -n "$items" ]
31             then
32                 items="\find $items -newer $time -print"
33             fi
34             ;;
35     esac
36     set - $items
37     ;;
38 esac
39
40 case "$#" in
41 0) exit ;;
42 esac
43
44 case "$mode" in
45 -n) echo "news: $*" ;;
46 -s) n="\echo $* | wc -w | tr -cd '[0-9]'"
47     case "$n" in
48     1) echo "1 news item." ;;
49     *) echo $n "news items." ;;
50     esac
51     ;;
52 *) start=
53     for item
54     do
55         if [ ! -r $item ]
56             then
57                 echo "news: Can't open /usr/news/$item" >&2
58             else
59                 echo "$start"
60                 start="\c"
61                 ls -l $item |
62 sed 's/[^ ]* *[0-9]* *\[([^\ ]*)\] *\[^\0-9\]*[0-9]* *\[\.*\] \([^\ ]*\)\$/\3 (\1) \2/'
63                 echo
64                 sed 's/^/ //' $item
65                 echo
66             fi
67     done
68     case "$time" in
69     "") ;;
70     *) touch $time && chmod go-w $time ;;
71     esac
72     ;;
73 esac

```

Checking Users' Disk Quotas

by Gilles Detillieux, Info West Inc.

An age-old problem with computer systems is running out of disk space. Even with the high capacity disks of today, the problem still exists; users' space requirements have no trouble keeping up with technology. When space gets low, you must get after the users to get them to clean up.

Often, all users need to keep from wasting excessive disk space is to be reminded from time to time, when their usage gets out of hand, that they should clean up their files. The shell script I am presenting here does just that. It can be run automatically, once per week, by cron. When run, it will mail a message to each user who exceeds his or her disk quota. This script can be useful on systems that do not already implement quotas by some other means.

The shell script, shown in listing 1, runs through the password file in search of entries where the fifth field ends with a number in parentheses. These numbers are taken as the disk quotas for those users, as a number of blocks. The fifth field (the GECOS field) is usually used for the user's full name. It can still be used for this purpose; the quota can be appended to this field.

The password file is run through "sed", to extract only the relevant information - the login name, quota and home directory of each user. The shell script will ignore entries which do not have the disk quota in parentheses at the end of the fifth field, because the substitution command fails, so "sed" will not print these. The extracted information is piped to the main loop of the script, which reads one line (one user entry) at a time. If a user's quota is set to 0, the script assumes no quota, and skips this user. Similarly, when the user's home directory is the root directory, or is not given, that user is skipped.

For each user that has a quota, the script runs "du -s" on the home

directory to get the total disk usage under that directory. If the total block count under that directory exceeds the user's disk block quota, then the script mails the user a message to that effect. It also prints a line on the standard output, indicating the culprit.

The procedure can be run automatically by placing the following line in root's crontab file:

```
0 3 * * 0 /etc/chkquotas
```

This will run the procedure every Sunday morning at 3 AM. You can run the procedure more or less often if you like, but once a week is about right. Early Sunday morning is a good time to run it because the system is usually inactive then, and users' will get their reminders when they get in Monday.

Because "cron" mails a user the output of his commands, the list of users exceeding their quotas, which the script sends to its standard output, will be mailed to "root" every week, so the system administrator can check up on these users. If your version of cron doesn't automatically mail the output of the script, you can explicitly pipe its output to "mail root".

Once the script is installed and the crontab entry is made, you will have to assign realistic quotas for each user, based on the amount of disk space you have, the number of users who must share this space, and the differing requirements of each user. You should then monitor the situation from week to week, to make sure that the users do clean up their files when they exceed their quota. When they don't, you may have to take steps to obtain their cooperation: teach them how to clean up and monitor their disk usage themselves, help them take inactive files offline, and if necessary, increase their quotas.

Listing 1: "/etc/chkquotas" shell script

```
#!/bin/sh
#
# chkquotas - check users' disk usage against their quotas
# - quotas taken from GECOS field of /etc/passwd
# (number in parentheses at end of field)
# - mails user a message if he/she exceeds quota
# - generates report on standard output
#

pw=/etc/passwd

sed -n \
's/:[^:]*:[^:]*:[^:]*:[^:]*:[^:]*\([0-9][0-9]*\)):\([^:]*\):[^:]*$/ \1
\2/p' \
$pw |
while read logname quota homedir
do
    if [ $quota -eq 0 -o "$homedir" = "/" \
        -o -z "$homedir" -o -z "$logname" ]
```

```
then
    continue
fi
usage='du -s $homedir | sed "s:$homedir: "'
if [ $usage -gt $quota ]
then
    # clean up numbers (remove space and/or
    leading zeros)
    quota='expr $quota + 0'
    usage='expr $usage + 0'
    echo "User: $logname,\tusage:
    $usage,\tquota: $quota."
    mail $logname <<!
    Your total disk usage under $homedir is $usage blocks.
    This exceeds your quota of $quota blocks.
    Please clean up your files.
    !
fi
done
```



Minutes From the Business Meeting January 10, 1989

1. Minutes:

MOTION : (Susan Zuk) The minutes from the January 10th, 1988 meeting be approved.

SECONDED : (Richard Wallacy)

In Favour : 15

Opoosed : 0

Carried

2. Membership Report:

At the present time we have 24 paid members.

3. Newsletter Report:

Thank you Kirk for your contribution to the newsletter. All contributions are greatly appreciated.

4. Treasurers Report:

A balance sheet may be included in the March Newsletter. At the present time we have a balance of \$230.00.

5. Miscellaneous:

There is a program on Channel 11 which promotes groups within the city. We should discuss the possibility of using this method of promoting our TUUG.

The Chamber of Commerce Business Show will be held April 26-29. We might be able to promote the group at members' booths.

Marty asked whether the group had discussed becoming affiliated with the International UNIX group /usr/group. Previously, to join /usr/group it required that there be 10 paid members of /usr/group. Memberships are \$100.00 per person. If people are interested in joining the International group or are members already, please let us know. We have been keeping this possibility in mind and will discuss it further as the group increases in size and as more members become interested.



Agenda
for
Tuesday, March 14, 1989
7:30pm
UNISYS
Canadian Indemnity Building
300-1661 Portage Avenue

- | | |
|----------------------------------|------|
| 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 | |
| 3. Break | 8:30 |
| 4. Presented Topic | 8:40 |
| System Administration Workshop | |
| a) System Accounting | |
| b) Security | |
| c) Backup/Restore Procedures | |
| d) Communications | |
| 5. Adjourn | 9:30 |