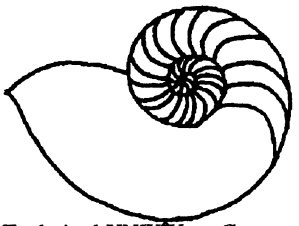

Technical UNIX® User Group

December 1989 $2.50
Volume 2, Number 3

newsletter of the Technical UNIX® User Group

This month ...

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

Thoughts From The Editor

By Susan Zuk

It's hard to believe that the Christmas season is here. I hope everyone is almost finished their Christmas shopping!! You haven't started??

This month's newsletter includes an important notice for all UUCP users and those members wanting to "make the connection" to the group's UNIX network. Our hub computer is making a name change. Make sure you change your site's Systems or L.sys files.

Our feature article discusses a different way of performing string searches. There are sample programs as well as performance tests done on the speed of the m4 utility compared to the awk and expr commands. Take a look and give the programs a try!!

Also included in this newsletter are two book reviews. For those who are working with or planning on getting into the X-window environment take a look at X Window Systems Programming and Applications With Xt. There also is a review on UNIX utilities for thoses users interested in increasing their vocabulary and knowledge of UNIX commands. What a great Christmas present for you!!

Kirk Marat has also submitted a shell script which will allow you to change devices stated in the /etc/inittab file. This script changes the respawn/off field. It is handy if you use one port to dial in and out but don't have the use of a bi-directional port. You might want ot put this script in cron if you tend to have dial-in facilities at night and dial-in during working hours. One good use right there.

Continue to read on and make sure you note the invitation at the back of the newsletter. Come one come all to the Christmas party. Bring along a friend who is interested but has never been to one of our regular meetings. It is an opportunity to meet the group and ask questions on a social level. See you next week and remember to R.S.V.P. Let me know by noon on December 11th so we can provide refreshments for all.

Bye for Now!!!

P.S. How about a New Year's resolution: Submitting at least one article in 1990?

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:	Matt Binnie	(W) 949-0190
Newsletter Editor:	Susan Zuk	(W) 788-7312
Membership Sec.:	Pat Macdonald	(W) 474-9870
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

ANNOUNCEMENT...

Meeting Location:
The December 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

Once again, I'm rushing to get this written at the last minute -just too many other things to do. Some day, I've got to get my life organized... naw! that would take much too long.

Since last month's newsletter, your humble president made a trip to Phoenix, Arizona, for the annual Neuroscience conference. It was much the same as last year's, in Toronto, except for the warm, sunny weather (sorry, I don't mean to gloat), and for the presence of even more Sun Microsystems than before - these seem to be a popular choice for medical applications. This was, in fact one of my predictions after last year's conference. The other, which didn't materialize, was the presence of NeXT Systems running medical applications. I expect that some will be seen next year, since they have been selling a lot of them to all kinds of academic types; it's just taking longer than I thought for the applications software to come. Some day, porting applications to new UNIX systems will be a lot simpler, once UNIX gets ITS life organized!

In addition to the hardware I saw at the conference, my new job at the University has given me the opportunity to see some of those new UNIX workstations. We had a Silicon Graphics Personal IRIS workstation (at more than $20K, I wouldn't call it personal, though) for a trial period, as well as a Sun SPARCstation I.

Sun and NeXT are so eager to sell to the University that both companies were out giving presentations on the same day! Needless to say, I was quite busy that day. It's really interesting to see all this, after having worked on business oriented UNIX systems with character terminals for the last few years. It's also good to see how active things are getting in the UNIX market, and to see another perspective on the computer industry than that given by DOS-coloured glasses.

Well, that's enough rambling for this month; on to business. The December 12th meeting will be a small pre-Christmas party, to get us into the festive mood, and to give us all a chance to chat informally (rather than having to all sit and listen to more of my rambling). The party will be held at the usual meeting location and time - Unisys, at 7:30PM. Please RSVP to Susan Zuk, so that we know how many people to expect in order to have the right amount of food and beverages.

We also have a some ideas for topics and speakers for meetings in the new year; we'll let you know as soon as we can confirm some of these. And as I said last month, if you have any ideas or wish-lists for topics and/or speakers, please bring them to the attention of any member of the executive.

I hope to see you all at the Christmas party. In any case, I wish you and your families and friends all the best for the upcoming holidays and the new year. Remember to drive safely, and watch out for reindeer!

NOTICE TO ALL UUCP USERS

I am writing this to all our regular UUCP connections, to inform them of our system name change. In order to enhance compatibility with other electronic mail systems, it was necessary to change our system name to one without uppercase characters. At the same time, we decided to change all system names on our local network to a set of names that is easier to remember. The former "UM535" is now called "larry". You may have to reflect this change in your UUCP configuration files to maintain proper operation of your uucp connection to our system.

Gilles Detillieux (larry!grdetil)

Hints to Help You Unleash the Power of UNIX

By Curt Motola and Bruce Stewart
Reprinted from /usr/group CommUNIXations November/December1989

UNIX users who frequently write shell scripts rely on utilities such as awk, tr, and expr to evluate expressions, extract substrings, determine the length of a string and to perform pattern substitutions. However, there is a single program that performs these and many other actions, often more simply and efficiently than its better-known counterparts. The program is m4, described by the UNIX manual as a macro preprocessor for Ratfor, C and other languages. In addition to this capability, m4 is also a useful shell programming tool, and features powerful built-in macro definitions and an intuitive syntax.

To demonstrate some of the power of m4, we've compared it with alternatives expr and awk on several shell programming tasks, Note that we are just using the built-in macro definitions of m4 and aren't taking advantage of its macro definition capability to accomplish any of the tasks used as examples. We rely only on m4's built-in macro definitions.

Figure 1 describes some of the more useful m4 built-in macros. The incr, decr, and eval macros provide general numerical processing capabilities, similar to those offered by expr. The eval macro in m4 includes functions unavailable in expr, such as bit operations and octal and hexadecimal number representation.

The other four macros provide useful string-handling functions, including the index function, which returns the position of the source string in the target string, or an error if the source string is not found in the target string.

Although both expr and awk offer many of these operations, m4 has an easier syntax than expr and is faster than awk. One reason for the performance difference between m4 and awk is the relative size of the executables; m4 is typically less than one-third the size of awk, and thus m4 loads more quickly and begins executing sooner. We attempted to get a rough measure of the relative efficiency of awk, expr, and m4 on the same task - extracting a substring from a larger string. To obtain our estimate, we created separate Bourne shell scripts for each utility. The first line of each script assigned a string to the variable $INPUT; the last line printed the value of the substring we wanted to extract. In between, the middle lines performed the extraction 100 times, using awk, expr or m4.

Figures 2 through 6 show an abbreviated form of the scripts used to compare performance. By timing the executions, we could estimate the relative performance of awk, expr and m4 for this task. Figure 7 presents the results of our tests.

Although we timed execution of the scripts on only one systemn, the results clearly favor m4 over awk when execution speed is a concern. The relative performance times for m4 and awk using a pipe (Figures 3 and 4), show that m4 is roughly 40 percent faster than awk for this syntax. The difference drops to about 10 percent when m4 uses a here document and awk uses /dev/null (Figures 2 and 5, respectively). Neither utility, however, approaches expr in execution speed.

We probably learned as much from writing the test scripts as we did by running them. Both the awk and expr versions require special characters for delimiters; often these must be quoted or escaped to avoid interpretation by the shell or the utility itself. The awk examples (Figures 4 and 5) also show the additional "print" and "END" keywords required to extract the substring. In contrast, the m4 examples feature a straightforward syntax. The m4 substring syntax is clearly more intuitive

Figure 1 - Useful m4 Built-in Macros

incr(arg)	Prints the value of arg incremented by 1.	index(arg1,arg2)	Prints the position in arg1 where arg2 begins (zero origin). Returns -1 when arg2 is not found in arg1.
decr(arg)	Prints the value of arg decremented by 1.		
eval(arg)	Prints the value of the arithmetic expression arg. You may use +,-,*,%,^ (exponentiation), bitwise &, |, ~, and ^ relationals and parentheses. Octal and hex can be specified as in C.	substr(arg1,arg2,arg3)	Prints the substring of arg1 starting at the position of arg2 for a length of arg3 characters.
		translit(arg1,arg2,arg3)	Transliterates the characters in arg1 from the set specified by arg2 to the corresponding character in arg3.
len(arg)	Prints the number of characters in arg.		

intuitive than expr's, which is based upon regular expression notation. And unlike awk, m4 preserves the C convention of indexing strings beginning at zero; the second character of a string is in position 1, just as it would be in an array in a C program.

Where m4 really shines, however, is in performing operations on multiple variables in a single invocation. Figure 8 provides one example of using several of m4's built-in macros (some in nested constructions) on multiple variables. Note that the example also uses the traceon macro, to display the results as expressions are evaluated.

Is m4 For You?

The examples presented here only begin to display the usefulness of m4. As we noted earlier, we've discussed only a few of the built-in m4 macros, those performing well-known functions such as substring extraction and arithmetic evluation. We haven't considered any of m4's more powlerful capabilities, such as macro definitions and output manipulation. Our intent was to offer an alternative to utilities such as awk and expr - an alternative that combines a straight forward syntax with reasonable execution speed. Is m4 for you? We encourage you to experiment, and to make the decision for yourself.

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 software consultant experienced in program development, systems administration and

Figure 2 - Using m4 With A Here Document To Extract A Substring

```
# m4   starts strings at index 0
       INPUT="NAME=William"
       PERSON=`m4 << -EOF
         substr( $INPUT,5 )
       EOF`
       echo Name of person = $PERSON
```

Figure 3 - Using m4 With A Pipe To Extract A Substring

```
       INPUT="NAME=William"
       PERSON=`echo "substr( $INPUT, 5)" | m4`
       echo "Name of person = $PERSON"
```

Figure 4 - Using awk With /dev/null To Extract A Substring

```
       INPUT="NAME=William"
       PERSON=`awk "END {print substr(\$"INPUT\",6)}" </dev/null`
       echo "Name of person = $PERSON"
```

Figure 5 - Using awk With A Pipe To Extract A Substring

```
       INPUT="NAME=William"
       PERSON=`echo "$INPUT" | awk "{print substr( $0, 6)}"`
       echo "Name of person = $PERSON"
```

Figure 6 - Using expr To Extract A Substring

```
       INPUT="NAME=William"
       A=`expr "$INPUT" :  ".....\(.*\)"`
       echo "Name of person = $PERSON"
```

Figure 7 - Relative Performance In Terms Of Expr

m4	using here document	(See Figure 2)	0.39
m4	using pipe	(See Figure 3)	0.48
awk	using pipe	(See Figure 4)	0.34
awk	using /dev/null	(See Figure 5)	0.35
expr		(See Figure 6)	1.00

Figure 8 - Using m4 To Perform Multiple Operations

```
COUNTER="12"
INPUT1="FIRST=Bill"
INPUT2="LAST=Barnes"
SEARCH="/lib/"
LOCATION="/usr/lib/font"
CITY="Oakland"

set `m4 << -EOF
traceon
incr($COUNTER)
decr($COUNTER)
eval(len(substr($INPUT1,len(FIRST=)))+len(substr($INPUT2,len(LAST=))))
substr($INPUT2,   len(LAST=),              1)
index($LOCATION,                           $SEARCH)
translit($CITY,     abcdefghijklomnopqrstuvwxyz, ABCDEFGHIJKLMNOPQRSTUVAXYZ)
EOF`

echo     COUNTER incremented by 1 = $1
echo     COUNTER decremented by 1 = $2
echo     The sum of the characters of first and last name = $3
echo     The first character of the person's last name = $4
echo     Is the search string found in the path (-1 = No)? $5
echo     The upshifted name of the city = $6
```

Book Reviews

Reprinted from /usr/group CommUNIXations November/December1989

X Window Systems Programming And Applications With Xt
(Prentice-Hall) Douglas Young, 480 pages; paperback

X Window Systems Programming And Applications With Xt provides an introduction to the X Window System along with some basic X Window terminology. The author is careful to point out that this is not a reference manual, but a tutorial. Each of the key features is illustrated along with working program explanations and screen pictures.

The book also discusses key concepts of the Xt Intrinsics layer. Several implementations of a simple X application are explored, including examples using handlers, callbacks and actions. The book also points out the advantages of using Xt Intrinsics and a widget set over using Xlib directly.

Widgets are discussed in depth, including using widgets to create a user interface and instructions telling programmers how to write new widgets.

Several chapters of the book discuss X Windows' graphics capabilities, discussing such subjects as graphics contexts and regions, Xlib functions used to display text in a window, the fonts X uses to represent characters, and the graphics facilities provided by Xlib and how they can be used with Intrinsics.

UNIX Power Utilities For Power Users
(MIS Press) John Muster, Peter Burns and Lurnix, 420 Pages; paperback

UNIX Power Utilities For Power Users is written as a set of Lurnix Modular Guides intended to assist the reader in mastering a broad variety of UNIX skills. The exercises teach the reader how to use a number of utilities, including many UNIX Power Utilities first described by Dennis Ritchie.

Each module begins with effective learning suggestions for users of all levels. Once a student's needs are determined, the modular concept is designed to quickly combine appropriate modules to create an effective training program focusing on the essential content for each student's special requirements.

The book takes a building block approach to UNIX utilities. The first few modules teach users such basic skills as how to issue commands to the shell to establish where utilities read their input and write their output; using the command line to pass information to utilities; and communicating with the shell using special characters.

Module four reviews the format for search and substitute commands. It also includes instructions for building regular expressions used in *grep, awk* and *sed.*

Concluding modules include a step-by-step examination of the cut and paste utilities; using relational files in a database; and locating files with *find.*

Lurnix is a group of professional educators and computer scientists who develop materials and deliver training both on-site and in their Berkley, CA, classroom.

Quick Tip

Quick Tip submitted by Kirk Marat

Here is the script for switching "devices" in the inittab file. They use the command name ($0) to decide whether the switch is from respawn to off, or from off to respawn. This requires having 2 links to the script, one named "ton" and the other "toff".

Program source:
Bruker Analytische Messtechnik GmbH
D-7512 Rheinstetten 4
Federal Republic of Germany

```
if [ $# = 0 ] ; then echo No argument ; exit 1; fi
if [ "`id | cut -d\( -f1`" != "uid=0" ]
then
  echo "You must be super user (root) to modify /etc/inittab"
  exit 1
fi
if [ $0 = toff ] ; then  command="/respawn/off/"; fi
if [ $0 = ton ] ; then  command="/off/respawn/"; fi
for arg in $*
do
   ex /etc/inittab << !
      /^$arg:
      s $command
      wq
!
done
init q
```

You Are Cordially Invited...

To Come and Rejoice
With Your Fellow TUUG
Members, Spouses & Friends,

To Welcome the Coming Holidays.

Tuesday, December 12, 1989
7:30 pm at Unisys
300-1661 Portage Avenue

❄❄❄❄❄

R.S.V.P.

Refreshments will be Served