February 1991 $2.50
Volume 3, Number 5

Technical UNIX User Group

newsletter of the

Technical UNIX® User Group

This month ...

Optical Disks: The Light Choice
A Helpful Shell Script
January Minutes
February Agenda

Late Breaking News...
Next Meeting to be held at the
UNIVERSITY of MANITOBA
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 use of Berkeley commands while in the Bourne shell environment. Kirk also mentioned that he can provide the script electronically if you are interested.

The second article deals with optical disks. This is an area which is receiving much interest. This technology allows the storage of huge amounts of data. The article states that over 640-6000 MB of data can be stored on a single disk. A big plus is that magnetic fields cannot harm the drives which is the case with magnetic storage media. A concern of the disks is that they are slower than magnetic tape drives. As with all new technologies, this is usually a time constraint and speed increases will be in future developments. For more information on optical disks take a look at the article.

The next meeting will be at the University of Manitoba. Our topic is on communications using sockets. This will be presented by Peter Graham. The agenda includes an abstract of what Peter will be discussing. Come on down to the meeting and hear what Peter has to say. Directions on how to find the location are below in the Announcement section. 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

ANNOUNCEMENT...

Meeting Location:

The February meeting location will be provided by The University of Manitoba's, Department of Computer Science, room 500A Machray Hall. This is the building that houses the Science Library and Dean's Office, and is the tallest, eastern-most building in the science complex, north-east of University Centre. Parking is available free in the evening in most lots. Access to the fifth floor of Machray Hall is likely to be restricted, so if the elevator won't take you there, get off on the fourth floor and take the stairs to the fifth, where we'll have the doors open for you.

A Helpful Shell Script, Just for Your Interest

Submitted by Kirk Marat

nshell simulates some C shell (BSD?) features for people who only have the Bourne shell.

```
# nshell 050490 Rudi Nunlist, UC Berkeley
# rnunlist@garnet.berkeley.edu
# rnunlist@californium.cchem.berkeley.edu
# Copyright (c) 1990 Rudi Nunlist.  May be freely
# distributed and copied
# for non-commercial use.
#
# csh -n or no argument makes new .history file
# csh -o keeps old history
# csh -d enables debug mode, keeps old history
#
#####################################################

VERS=050490.1
export num
HISTFILE=$HOME/.history

set_ps3()
{
PS3="`pwd`"
}

add_hist()
{
 echo " $num  $CMD   " >> $HISTFILE &
 LCMD=$CMD
 num="`expr $num + 1`"
}

echo "       nshell $VERS"

if [ "$num" != "" ]
then     num="`expr $num + 1`"
else num=1
fi

PS1="$LOGNAME"
```

```
set_ps3

if [ -s "$HOME/.nshellrc" ]
then   . $HOME/.nshellrc
fi


if  [ $# != 0 ]
then      case "$1" in
          -o)
               if [ ! -s "$HISTFILE" ]
               then      echo " history file was empty or
not found"
               else Z=`wc -l $HISTFILE | cut -c5-7`
                    num=`expr $Z`
                    if [ "$num" = "0" ]
                    then      num=1
                         echo " history file was empty"
                    else LCMD=`tail -1 $HISTFILE |cut
-c5-`
                    fi
               fi;;

          -n)  cp /dev/null $HISTFILE ;;
          -d)  set -x;;
          *)   echo "Usage: nshell [-o] [-n] [-d]"
               exit 0;;
     esac
else cp /dev/null $HISTFILE &
fi

trap '' 15
trap : 2
trap 'exec nshell -o' 0
#PS1="$LOGNAME"
#set_ps3

#if [ -s "$HOME/.nshellrc" ]
#then   . $HOME/.nshellrc
#fi
```

```
while :
do
    echo "$PS1$PS3> \c"
    read CMD
    case "$CMD" in
        '') continue;;
        \!) continue;;
        h)  if [ `expr $num` -gt 22 ]
            then
                pg -p " Page %d Press <space> for more, <-> for\
prev. page, q to quit" -nc +`expr $num - 20` $HISTFILE
            else cat -u $HISTFILE
            fi;;

        !*) if [ "$CMD" = "!!" ]
            then    CMD=$LCMD
            else Z="`echo $CMD | cut -c1-2`"
                if [ "$Z" = "!!" ]
                then    Z="$LCMD`echo $CMD | cut -c3-`"
                    CMD=$Z
                    if [ "$CMD" != " " ]
                    then add_hist
                    fi
                else Z="`echo $CMD | cut -c2-5`"
                    if [ "`echo $Z |cut -c1-1`" = "-" ]
                    then Z="`expr $num + $Z`"
                    fi
                    CMD=`grep " $Z " $HISTFILE |cut -c5-`
                fi
            fi
            echo $CMD
            if (eval $CMD)
            then    case "$CMD" in
                    cd\ *)
                        eval $CMD
                        set_ps3
                        LCMD=$CMD
                        continue;;
                esac
            fi
    LCMD=$CMD;;

        \^*\^*) Z=`echo $CMD | tr "^" "/"`
            ZZ=`echo $LCMD | sed "s$Z/" `
            if [ "$ZZ" = "" ]
            then    echo "    $CMD :    substitution failed !"
            else CMD=$ZZ
                echo $CMD
                add_hist
                if (eval $CMD)
                then    case "$CMD" in
                    cd\ *)
                        $CMD #(eval $CMD)
                        set_ps3;;
                    esac
                fi
            fi;;
        exit)   trap 'exit' 0
            exit 0;;

        *=*)    add_hist
            eval $CMD;;

        *)    add_hist
            if (eval $CMD)
            then
                case "$CMD" in
                cd) cd
                    set_ps3
                    continue;;
                cd\ *)  eval $CMD
                    set_ps3
                    continue;;
                export\ *)
                    $CMD
                    continue;;
                esac
            else :
            fi;;
    esac
done
```

Optical Disks: The Light Choice

By Jennifer E. Beaver
Reprinted from /usr/group CommUNIXations February1991

Computer systems have voracious appetites. Everyone wants more speed, more memory and greater storage capacity. UNIX systems are no exception. As they expand into larger user environments, and applications and data require more memory, the need increases for high-capacity, easily accessed storage devices. Traditional magnetic media, even very large hard disks, soon top out when faced with uses such as company databases, multimedia applications, collections of complex graphics and video. Optical disks are becoming a solid solution to this problem.

Optical disks are efficient and cost-effective. A single CD-ROM, identical in size and appearance to an audio compact disk, stores more than 640MB of data at a cost of less than $15 per disk.

Though well suited for storing space-hogging images, optical storage devices are used primarily for archiving data. Their ability to store text and graphics inexpensively and conveniently makes optical disks popular for archiving. Medical records stored on optical media, for example, can depict X-rays and handwritten notes as well as typed text.

Optical technology offers clear-cut advantages over paper documents, which are easily misfiled and often difficult to retrieve. These disks save a tremendous amount of physical space; one can store thousands of pages of easily retrievable text. In addition, some optical disks provide an audit trail crucial for sensitive documents.

OPTICAL UPS AND DOWNS

Optical disks rely on a high-powered laser beam to write to disk and a less powerful laser to read from it. Unlike traditional magnetic media, the laser never touches the storage platter. As a result, optical storage suffers less erosion of data. Magnetic fields don't harm optical drives, which also makes them more rugged than their magnetic storage counterparts.

As already suggested, the biggest advantage of optical disks is their capacity. One 940MB, 5.25-inch optical disk can store as much information as 20 four-drawer file cabinets -- the equivalent of 500,000 pages of text, which would require 780 floppy disks. In addition, the 12-inch or 5.25-inch disks are often stacked in jukeboxes, which allow millions of text pages to reside in one place.

For archiving, optical storage provides a distinct advantage over tape because data can be accessed randomly. Most optical storage management software makes the optical system appear like familiar magnetic media, which simplifies storing and retrieving data.

However, the advantages of optical disks are offset by their slower access speed. Unlike a constantly spinning magnetic disk, an optical disk stops when the laser reads or writes to it. Some developers are compensating for the media's natural lag time by incorporating SCSI connectors that increase a system's overall throughput (see "Optimizing The Interface").

Because optical drives weren't around when UNIX was developed, the operating system doesn't recognize them, requiring a software driver that makes UNIX believe it is interfacing with familiar magnetic media. And since traditional UNIX requires backing up off line, vendors who offer optical products for on line transaction processing applications have had to find ways to circumvent this time-consuming process. Vendors have taken a variety of approaches to integrating optical storage with UNIX, including complete subsystems with integrated hardware, software and firmware. Because the original UNIX developers didn't anticipate the vast quantities of data the operating system would one day be asked to handle, its interface with optical media remains somewhat tricky. This problem is shared by other operating systems developed before optical systems came along.

ERASE OR DON'T

Optical disks first appeared as Write Once Read Many (WORM) systems, which offer a permanent record of all data recorded. The data cannot be erased, which is a valuable feature for some applications and a roadblock to others. Insurance companies, government agencies and medical facilities all demand long-term, unalterable document storage, so unerasable WORM media affords the protection they need.

The recent introduction of erasable optical is increasing the media's popularity. Rewritable optical products are emerging, and observers believe that they will give the market a shot in the arm. For document creation applications such as CAD/CAM, desktop publishing and multimedia production, where material is more volatile, erasable optical offers flexibility. Though market projections for WORM products have been positive, some industry insiders find them overly optimistic. "The optical market hasn't taken off like we expected," says Timothy Green, president of N/Hance Systems. On the horizon are multifunction products that incorporate both permanent and erasable technologies.

INTEGRATED SOLUTIONS

Sometimes the best storage solution isn't one technology but several. That's the philosophy of Epoch Systems, which offers a dynamic management system for hierarchical storage on a workstation network. Frequently accessed data are kept on magnetic media while other information resides on optical disks. Users don't know or care where their data actually resides. To them, the storage system appears like a Winchester hard disk drive with unlimited capacity.

This philosophy makes sense to the Xerox Palo Alto Research Center (PARC) System Sciences Laboratory in Palo Alto, CA, which uses an Epoch-1 Infinite Storage Server to accommodate the nearly 100 scientist and researchers on its network of Sun Microsystems computers. The lab's data storage and management problems began when a switch from proprietary Xerox workstation to Suns opened the door to the wealth of UNIX software. As users bought and installed applications, the server disk filled up rapidly, which made file movement and directory reorganization necessary uncomfortably often. The network now has 8MB of main memory, 2.3GB of magnetic disk storage and 28.8GB of optical disk storage.

Despite the fact that optical storage is slower than magnetic storage, the worst case for PARC network users is 11 seconds to access data stored on an optical disk. On the other hand, finding and mounting a tape for backup could take hours.

QStar Technologies also offers a hierarchical storage management system that integrates RAM with magnetic and optical systems. Called QStore, the software prioritizes data files and places frequently accessed material on magnetic media and less-often-used data on optical. This product also makes the whole storage setup appear to users like a Winchester disk with unlimited capacity. The American Chemical Society at Cornell University in Ithaca, NY, enlists QStore to manage abstracts of chemical processes in scientific papers. The software automatically determines whether to place data on magnetic or optical media, according to frequency of access.

MODULARITY FOR PORTABILITY

QStar chose a modular approach to its software offerings that allows unusually fast portability to various platforms. Unlike other file system software that fools the operating system into working with optical products and may require modification of the operating system itself, the QStar line takes advantage of the UNIX file system and uses standard UNIX system calls and utilities. Since they work so closely with the native operating system, the software modules require only a 5 percent alteration in source code to run on new platforms.

Designed for UNIX-based workstations, the QStar products rely on Network File System (NFS) as their networking medium. The WORM

Optical File System (OFS) Storage Software for 12-inch drives and autochanger subsystems appears to users as magnetic media. Woodland Geophysical of Woodland, TX, uses OFS to prepare optical drivers packed with seismic information for its customers in the oil industry. The company massages and reformats the data, initially recorded on tape, and records the streamlined result onto 12-inch optical platters. According to Richard Koseluk, vice president, each project generates about 100 tapes, which translates into one 12-inch optical disk with 1GB to 1.5GB of data.

Woodland Geophysical also archives its own data using OFS, which allows users to randomly access its nearly 100 optical disks. OFS replaced software that caused data integrity problems with faulty writing to the WORM platter.

QStar also offers Magneto-Optical Storage (MOS) Systems Software, which operates identically to OFS but is engineered for rewritable systems. Another software product, an intelligent autochanger called ViewStore, manages the robotic control functions that move platters from shelves to drives as needed. It can adjust itself based on user loads and data access patterns, though the software can also be tailored by system administrators.

OPTICAL DATA SWAPPING

Some users have another concern in addition to storage capacity and access time. They must also share data with users of different operating systems.

Several software solutions allow data swapping with optical media. Write Once File System (WOFS) software for WORM from N/Hance Systems lets users exchange optical disk cartridge among WOFS-equipped computers for different manufacturers running different operating systems by formatting the data the same way on each optical disk. Now available for the PC-DOS, MS-DOS and SCO UNIX/386 2.3 operating systems, WOFS will soon support systems from Apple Computer, Sun Microsystems, Digital Equipment Corporation and Novell and also run under OS/2.

The software relies on an independent file structure for the control of optical and removable media. Within the WOFS file structure, redundant transition checks enhance WOFS' probability for data recovery, according to Timothy Green. Interspersing data and directories on the disk has led to faster access time.

Because it also allows data interchange with erasable optical drives, OSS WORMware from Optical Software Solutions is something of a misnomer. Available for MS-DOS, Sun NFS and Novell Netware systems, WORMware supports optical drives ranging in size from 5.25 to 12 inches and capacities from 600MB to 6,000MB per disk. Like WOFS, WORMware features an independent file system and appears to users like a familiar magnetic disk storage system.

Image Resources of Santa Clara, CA, which converts documents on film and microfiche into more readily accessible optical storage on jukeboxes, chose WORMware to play a part in its project for the San Benito Land Title Corporation, a property title company in Hollister, CA. In converting 250,000 land title records, Image Resources used WORMware to enable an MS-DOS application and a jukebox with a WORM drive for communications and storage.

As UNIX increases in popularity, its appetite for storage will continue to grow. In addition to multifunction optical products, which allow both WORM and erasable technologies on the same storage media, the future holds more streamlined products that will better tap the strengths of UNIX while making up for its difficulties in recognizing optical disks. Eventually, new developments may make optical drives comparable in speed to magnetic drives. Users and developers may also discover that the storage solution of the future is not a single technology but several integrated ones.

__Jennifer E. Beaver__ is a free-lance writer based in San Pedro, CA. Her work has appeared in a variety of high-technology publications.

Technical UNIX® User Group

TUUG Business Meeting Minutes

Tuesday, January 8, 1991

Attendance:

7 members, 4 guests present.

Minutes of November meeting:

Reviewed and accepted.

Membership Secretary's Remarks:

Membership fees are $20 per year, pro-rated, and are due in October of each year. They are not subject to G.S.T.

Newsletter Editor's Remarks:

We are always in need of more articles for the newsletter. Please contact Susan Zuk if you have anything to submit. Kevin McGregor, who gave a presentation on Coherent, suggested that any update information he receives about Coherent could be forwarded to the group.

Treasurer's Remarks:

Expenses for the Christmas party were just over $50. Postage expenses in 1991 will be slightly higher, due to both a small increase in postal rates and the G.S.T., but this increase will be easily covered by current funds.

Meeting Coordinator's Remarks:

Future meeting topics are in the works for the rest of the business year, until the summer. Possible topics to be covered are socket programming, PC-NFS, SCO Unix with Open Desktop, and a presentation by HP on their strategy with respect to OSF, products, etc.

Technical UNIX® User Group

Agenda

for
Tuesday, February 12, 1991
7:30pm
University of Manitoba
Room 500A, Machray Hall
(See Page 2 - Announcements for Details)

1. Round Table	7:30
2. Business Meeting	8:00
a) Minutes of January's Meeting	
b) Membership Secretary's Report	
c) Newsletter Report	
d) Treasurer's Report	
4. Break	8:30
5. Presented Topic	8:40
Inter-Process Communications Using Sockets	
Presented By Peter Graham	
6. Adjourn	9:30

Inter-Process Communications using Sockets

An Abstract

By Peter Graham

Modern software is often structured as a group of cooperating processes. In order to permit this cooperation, it is necessary for the processes to communicate with one another. Inter-Process Communication (I.P.C.) is supported in Berkeley Unix by a facility known as sockets. Socket primitives permit processes on the same or different (networked) machines to talk to one another easily and efficiently. This talk will give a programmer's view of the socket system calls and will include the development of a simple example.