Establishing and Administering a Public Access Internet facility

Sheri Byrne
Gemini Learning Center

Establishing and Administering a Public Access Internet facility

Sheri Byrne – Gemini Learning Center

ABSTRACT

In a perfect world, Internet access should be available to anyone who wants it. In reality, that is not an easily achievable goal. In many geographical locations there are "haves" – people with Internet access via large corporations or academic connections and "have nots" – contractors, consultants, and employees of small companies with no way of accessing the Internet.

The Calgary Unix Users Group (CUUG) Computer Resource Centre (CRC) was established in March 1991 to provide UNIX facilities and Internet connections to its members. This paper describes the evolutionary process that went into the conception, establishment, and current operation of the CUUG CRC (Internet address cuug.ab.ca). This includes both technical issues such as physical administration of the systems, and non-technical (but equally important) issues such as coordinating volunteer and fundraising activities and legal ramifications.

Motivation

Calgary, Alberta has the third largest base of UNIX workstations in North America (DMR, 1988), yet most Albertans had little or no access to the Internet prior to 1991. All access to the Internet in the province of Alberta is controlled through the Alberta Research Council (ARC), a government-subsidized, high-technology corporation, who charge a substantial yearly fee for corporate Internet access, significantly less for non-profit access. The only other method of Calgary Internet access in 1989 without incurring long distance charges was through the University of Calgary, which at that time had limited non-student access, which has since been suspended.

Some Calgary-based uucp feeds were available which supported mail and netnews access. However, these were considered to be too "home-grown" for any type of serious usage. Phone line access was limited, down time was considerable, and the majority of these establishments were run out of basements as hobbies and bulletin boards rather than as full-time, business-oriented enterprises.

The need for dissemination of UNIX information was evidenced by the establishment of the Calgary UNIX Users' Group in September, 1990. Less than three months after it began, CUUG boasted a membership of over 300. CUUG's primary goal is to encourage the use of UNIX and open systems. In January, 1992 the CUUG Membership Services Committee decided to partially fulfill this goal by providing Internet access to its membership.

Scope

The CUUG CRC currently consists of five ethernetted UNIX platforms, an Intel-based 486 which is owned by CUUG, the others (IBM RS6000 320, Silicon Graphics Indigo, HP 9000/730 and Sun Sparc 2) on loan from hardware vendors (see Figure 1 for network layout details). The system currently has fourteen telephone lines. Two of these lines are dedicated to the University of Calgary, one line is unlisted for administrator use only, and eleven lines are for public access. There is a Gandalf terminal server on the front end to transfer login access to the specified UNIX system via TCP-IP. The CRC performs the following functions:

1) Access to Internet e-mail, ftp, telnet, net news
2) Archive site for exchange of newsletters amongst UNIX users group
3) Application porting facility for both non-UNIX and UNIX software programs to other UNIX platforms.

The CUUG CRC had to be solid enough for long-term, business-oriented projects like software ports. However, the setup costs of $8900 CDN and initial yearly budget of $11,000 CDN did not provide resources for a paid system administration staff of any kind.

Hardware Logistics

Originally, many people thought that acquiring the hardware necessary to run such a system would be a major stumbling point. However, this turned out to be the least of the logistical problems. This was accomplished by tying the CRC hardware with the CUUG Corporate Sponsorship program. In the first year, only Gold level sponsors were allowed to loan hardware to the CRC, in succeeding years, hardware loan is now a requirement for Gold Sponsorship status.

It was determined that a long-term loan, rather than donation, was the preferable method of obtaining the hardware. This bypassed potential problems and costs associated with insurance and service contracts, since the sponsors still owned the systems.

Since its inception, CUUG has had a waiting list for gold corporate sponsor status. However, to guard against the day CUUG did not have such a luxury, a 486 running Interactive Unix was purchased, to guarantee that there would always be at least one server available to provide Internet access.

One of the side-effects of the "hardware loan" arrangement is system volatility. In addition to the six hardware vendors currently represented, hardware from DEC, Bull and AT & T has also been part of the CRC over the past 2 1/2 years. There have been several occasions where systems have been pulled or swapped on very short notice due to the vendors desire to upgrade to a better system, or need to sell an asset to avoid tax implications. Therefore, CRC administrators have been forced to make many decisions with this hardware volatility in mind.

One decision influenced by this is the need to make only full filesystem dumps, for quick disk restoration. It was not feasible to obtain an 8mm or DAT tape drive from every sponsor, therefore, CUUG's second major hardware investment was an Exabyte 8mm tape drive which was put on the CUUG-owned 486 and networked to the other systems.

Location Logistics

The CRC had to be located in a vendor-neutral setting which was easily accessible by the members who wished to do on-site work which required X Window Systems support, such as graphical user interface development. This ruled out vendors themselves, VAR's of vendors, and anyone who wasn't in the downtown core, where the majority of Calgary computer-oriented business is conducted. In addition, the space had to be free, and people who knew rudimentary UNIX system operation had to be available in case of a system emergency.

The CUUG CRC hardware was eventually installed at the offices of Gemini Learning Systems, Inc. Five isolated ground circuits and five telephone lines (eventually expanded to fourteen) were added to support CRC system operation requirements. The CUUG CRC systems are available 24 hours/day, 365 days a year via modems ranging from 2400 baud to 14.4 Kbaud, and during normal weekday business hours.

Administration Personnel

The logistics of adminstering the hundreds of users on the five CUUG CRC systems was compounded by the fact that it was done on an entirely volunteer basis. System setup and administration tasks were broken down into the following categories:

- News
- Mail
- UUCP feeds
- Software Installation
- Account Maintenance
- Administration which required physical system access (backups, hardware troubleshooting, manual system reboot, etc.)

and assigned to five different volunteers. Those who absolutely required root access were provided it; all others were added to the "helpers" group where they had privileges to install software. Any system work which required root access or rebooting was coordinated through the chairperson of the CUUG Membership Services Committee to avoid potential system conflicts.

Users were advised to send questions and requests to the postmaster, who forwarded them to the appropriate volunteer. A voice telephone line answered by a non-technical person was available for general system information and to report emergencies such as system crashes or modem outages.

The efforts of the five original volunteers were augmented by a "System Administrators Assistant" work / co-operative education program established between CUUG and the Southern Alberta Institute of Technology (SAIT) Computer Technology Department. Every semester, two SAIT students would work at the CUUG CRC doing day-to-day system administration tasks such as account maintenance and backups. Each student was responsible for developing an application sometime during the semester which automated part of the system administration process which was left as a legacy to future students and other CUUG volunteers.

Security

System administration tools (such as "cops" and "crack") were obtained and run regularly as a first line of security defense. In addition, several shell scripts were developed to aid in monitorng CUUG's unique requirements. These scripts were developed in Bourne Shell for portability.

It was illegal to possess any password cracking utilities on the CRC without first notifying one of the administrators that the files would be there temporarily, and notifying the system administrator of the destination machine they were eventually being transfered to. Anonymous ftp was restricted to one system only.

All attempts to "su" were tracked (see program "intruders"). Actual system usage at a process level was also tracked (see program "monitor") in order to generate reports to the hardware sponsors to let them know of both the quantity and nature of the traffic on their particular system.

Disk Management

CUUG wanted to provide the most flexible environment possible for their users. There is approximately 1.6 GB of disk space across the five systems for the users to share. After much discussion by the CUUG CRC volunteer administrators, it was decided to manage this disk space without implementing fixed quotas. In order to do so, it was necessary to be able to monitor where the disk space was being used, by user, and most importantly on an incremental basis (i.e. which users had recently used a large chunk of space)

A disk space monitoring program which reviewed current usage against the most recent report separated reported on who was using more than 1 MB of disk space (see program "disk_pig"). An adjunct program sent mail automatically (see program "hog_mail") to both root and the offending user. Users who requested disk space leniency for a particular project for a fixed period of time were excluded from the disk space policing, but their usage was monitored through quota.

When disk space got tight, there were checks made for multiple copies of common utilities like kermit or xdbx. The duplicated copies were compressed, the users were pointed to the system copy, and they were asked to remove their private copy. Core files were removed nightly, and accounts where the membership had lapsed were deactivated (see programs "starout", "deactivate", "delete_users") removed after several warnings and two months. Core files were deleted nightly (see program "corekiller").

After two years, due to the overwhelming amount of e-mail, a 100K limit was placed on both outgoing and incoming e-mail.

Shared logins, Use Limitation, and UUCP feeds

Because CUUG is a non-profit organization, the volunteer administrators discouraged users from sharing accounts with others in their companies. With only four public access lines in the initial configuration, it would not have been fair for one login to be re-used on all four lines, effectively blocking other equal paying members from using the system.

This was originally established as a recommendation, and eventually after some abuse had been discovered, was turned into a rule. A shell script was written (see program "countuser") to record multiple simultaneous logins from the same account, a second was written to terminate connections after two hours.

The CUUG CRC was established to assist companies in their open system business computing requirements, not substitute permanently for inadequate computing facilities or to actually run a business. Therefore, CUUG CRC use is limited to twenty hours per week (in person) and six hours per day (in two hour segments) by phone.

Companies with more than five accounts were encouraged to establish a UUCP feed to obtain their news and mail, rather than clogging the lines with individual access. Times were allocated and dedicated to these companies for UUCP access.

Anonymous FTP, FSF Software Distribution

CUUG purchased a full set of FSF tapes for gcc, emacs, and the X Window System, and distributed these materials throughout Southern Alberta. Members were encouraged to use the actual tapes when possible.

An anonymous ftp site called anonymous@cuug486.cuug.ab.ca was established for the exchange of newsletter articles and transcripts from open systems presentations between CUUG and other groups like CUUG around the world.

Non-Technical System Administration Issues

Many issues pertaining to public access Internet administration have no direct link with technology, but are equally problematic as the technical issues. Many of these "non-technical" issues cross over into the technical world when a business decision, such as "CUUG members may not share their accounts" results in having to write a program which enforces that decision.

Legal protection was the largest of the non-technical system issues. A CUUG CRC usage agreement was developed, and had to be signed and the original on file at the CUUG office to ensure that CUUG was not liable for any financial or business loss associated with work done on the CRC, and that the CRC was protected if CRC users were conducting illegal activities.

Other legal issues involved careful investigation of the somewhat contradictory rulings of the Royal Canadian Mounted Police (RCMP) and Canadian Radio and TeleCommunications (CRTC) with regards to some of materials distributed in the rec.* and alt.* news groups.

The CUUG CRC currently has a budget of approximately $21,000 CDN per year to cover the current level of services, which includes the costs of connecting to the Internet, telephone lines, equipment lease, and miscellaneous costs such as backup tapes. CUUG membership is $100 per year, with corporate discounts available. The remainder of the

the costs of operating the CRC come from CUUG corporate sponsorships.

Conclusions

The presence of the CUUG CRC has made a large impact in Calgary, especially in the area of small business. There have been the instantaneous benefits of access to e-mail, ftp, archie, gopher. Many companies utilizing the CUUG CRC have reported that they are now able to discuss problems and opportunities with a much larger group reflecting more international concerns than every before. Many people have used the CUUG CRC to teach themselves UNIX, or learn a new UNIX platform without cost in classes or hardware acquisitions. Small software companies have been able to port applications more easily, and everyone has been able to evaluate more combinations of hardware and software than they would before the CRC came into existence.

Internet access should be available to anyone who wants it at a reasonably affordable cost. A computer environment such as the CUUG CRC is one way of providing this kind of access.

Author Information

Sheri Byrne has a B.Sc. in Information System Management from the University of San Francisco. She has worked in the field of UNIX and open systems for the past 9 years, the last 4 at Gemini Learning Systems in Calgary, Alberta, Canada, where she is Director of Technical Operations. Sheri was a founding member of the Calgary UNIX Users Group, and was a member of the board for two terms. She can be reached by mail at 1750, 101-6th Avenue S.W. Calgary, AB, Canada, T2P 3P4, or electronically at byrne@posc.org until November 29, 1993, and byrnes@cuug.ab.ca after that.

APPENDIX

Intruders

```
# Make a log of all users who have tried to "su" another account.

olddate=‘grep "sun" intruders.out | cut -c1-15 | tail -1‘
echo $olddate
date >> intruders.out
echo "---------------------------------------------" >> intruders.out
grep "su:" /usr/adm/messages | grep -v "byrnes" | grep -v "ingoldsb" >> intruders.out
echo "---------------------------------------------" >> intruders.out
echo " " >> intruders.out
echo " " >> intruders.out
```

Monitor

```
# This program is designed to monitor users logged into the system.
# This program is executed intentionaly as an endless loop. It checks
# for logged in users.  If no users are on the system, the process
# sleeps for about 30 seconds and then wakes up and checks the
# system for users again.  Once logins are detected, the users are
# monitored until they have all logged out and then repeats the
# sleep sequence.

while [ $-z $done ]
do
  list=‘users‘
#

  if [ -z "$list" ]
  then
     sleep 30

#
  else
     echo "" >>userlogs
     w | grep -v ’sleep 300’ >> userlogs
     echo "" >>userlogs
     sleep 30
  fi
#
```

```
# Writes the user information into a file called "userlog"
#
# A report is generated at midnight using all the information
# collected by the userlog. The userlog is then emptied for
# further use by the process.
# The name of the reports are generated by taking the system
# time and cuting the date from it (e.g. the name should look like
# this Jul17.rpt).

time=`date | cut -c12-19`

if [ "$time" -eq "00:00:00" ]
  then
   month=`date | cut -f2 -d" "`
   day1=`date | cut -c9`
   day2=`date | cut -c10`

   if [ "$day1" -eq " " ]
     then
       day1=0
   fi

   filename=$month$day1$day2.rpt
   cat userlogs >> $filename
   echo "">userlogs
fi
done
```

Deactivate

```
# Program runs before system login and to either call system login
# program or not.

# This program will accept user’s login name, check to see if it exists
# in the file called ’deactive.tmp’.  This file contains the login
# names and renewal dates for selected users.  Appropriate messages as
# well as either allowing the login to proceed or being denied login
# permission occur in this program.

echo "Type your logname."
read logname
match1=`grep $logname /users/cuug/deactive.tmp`
echo $match1
count=0
for name in $match1
do
   if [ "$name" = "$logname" ]
   then
      match=$name
      count=1
      echo "$match is match $count"
   elif [ "$count" -ge 1 -a "$count" -le 3 ]
   then
      match=`echo $match $name`
      count=`expr $count + 1`
      echo "$match is match $count."
   fi
done
echo "In deactive file as $match."

# Check to see if there is a match in the deactive.tmp file or not.
# If no match, let user login.  If there is a match, set variables
# and continue with validity checks.
```

```
if [ -z "$match" ]
then
#    sh /bin/login.unix
    echo "no match."
    exit
else
    deleteyear=‘echo $match | cut -f4 -d" "‘
    deletemonth=‘echo $match | cut -f2 -d" "‘
    deleteday=‘echo $match | cut -f3 -d" "‘
    currentyear=‘date ’+ %y’‘
    currentmonth=‘date ’+ %m’‘
    today=‘date ’+ %d’‘
fi

# If there is a match, send message and do appropriate action as
# described in the following program segment.

if [ "$deleteyear" -eq "$currentyear" ]
then
    if [ "$deletemonth" -eq "$currentmonth" ]
    then
        if [ "$deleteday" -lt "$today" ]
        then
            echo "Membership expired on $deletemonth/$deleteday/$deleteyear."
            echo "Contact Janet at 265-2289 for renewal."
            exit
        elif [ "$deleteday" -eq "$today" ]
        then
            echo "Your membership expires today.  Contact Janet at 265-2289"
            echo "for renewal."
#            sh /bin/login.unix
            exit
        elif [ "$deleteday" -gt "$today" ]
        then
            echo "Membership expires on $deletemonth/$deleteday/$deleteyear."
            echo "Contact Janet at 265-2289 for renewal before this date. "
#            sh /bin/login.unix
            exit
        fi
    elif [ "$deletemonth" -lt "$currentmonth" ]
    then
        echo "Membership expired on $deletemonth/$deleteday/$deleteyear."
        echo "Contact Janet at 265-2289 to reinstate. "
        exit
    else
        echo "Membership expires on $deletemonth/$deleteday/$deleteyear."
        echo "Contact Janet at 265-2289 for renewal before this date. "
#        sh /bin/login.uni
    fi
elif [ "$deleteyear" -gt "$currentyear" ]
then
    echo "Membership expires on $deletemonth/$deleteday/$deleteyear."
    echo "Contact Janet at 265-2289 for renewal before this date."
#    sh /bin/login.unix
else
    echo "Membership expired on $deletemonth/$deleteday/$deleteyear."
    echo "Contact Janet at 265-2289 to reinstate."
fi
exit
```

Starout

```
# Script to replace the "!" in the passwd file with a "*" to prevent a
# user from logging on. Also enters the user into the deleted.users file.

# find the user
cp /etc/passwd /etc/pass.sav
continue=’y’
clear
echo " "
while [ "$continue" = ’y’ ]
do
   echo " "
   echo "   Enter user to delete : "\c
   read user
   echo " "
   usrline=`grep -n -i $user /etc/passwd`
   usr=`echo $usrline | cut -f6 -d":"`
   lne=`echo $usrline | cut -f1 -d":"`
   zap=’N’
   if [ "$usr" ]
   then
      echo "      Remove user "\c
      echo $usr "(y/n)  "\c
      read zap
      echo " "
      if [ $zap = ’y’ ]
      then
          sed """$lne s/!/*/""" /etc/passwd > /etc/delusrtemp
          mv /etc/delusrtemp /etc/passwd
      else
          echo "  Deletion Terminated !"
      fi
   else
      echo "      User not found ! , Press Enter "
      read jnk
   fi
   echo " "
   echo "   Continue Deletions ? (y/n) "\c
read continue
done
grep -n "\*" /etc/passwd | cut -f1,2,6 -d: > /etc/deleted.users
```

Corekiller

```
# shell program to find and remove all core files.

# core files are the saved images of programs that crash. these files
# can be extremely large so it is necessary to periodically seek out
# and destroy them.

coreusr=`find / -name core -print -exec rm -f {} \; | \
                     cut -f3 -d"/" | tee cores.usr`
echo $coreusr

for cusr in $coreusr
do
      qusr=`grep $cusr /etc/passwd | cut -f1 -d":"`
      if [ "$qusr" = "$cusr" ]
      then
            mail $cusr < cleanup.mes
      fi
done
```

Delete_users

```
# This file is run two months after CUUG membership expiration to remove
# obsolete accounts

cp /etc/passwd foo

tail +13 foo | grep ":\*:" | grep -v fsf | cut -f6 -d":" > dirs_to_hose

for file in ‘cat dirs_to_hose‘
do
   echo $file
   echo -n "Do you want to delete this guy?"
   read answer
   if [ "$answer" = "y" ]
   then
      rm -rf $file
      echo $file deleted
   else
      echo $file NOT deleted
   fi
done
```

Countuser

```
# Count how many times each user is currently logged into system and, if more
# than one, send user and root messages via electronic mail to inform them of
# time, date and number of logins.

howmany=‘who|wc -l‘
noduplic=‘who|cut -c1-8|sort -u|wc -l‘
if [ "$howmany" -eq "$noduplic" ]
then
    exit
else
    unset howmany
    unset noduplic
    howmany=‘who|cut -c1-8‘
    noduplic=‘who|cut -c1-8|sort -u‘
    for activename in $noduplic
    do
        count=0
        for name in $howmany
        do
            if [ "$name" = "$activename" ]
            then
                count=‘expr $count + 1‘
            fi
        done
        if [ "$count" -gt 1 ]
        then
            echo "‘date|cut -c1-10‘">badlogin
            echo $activename>>badlogin
            echo "Currently have $count logins">>badlogin
            echo "More than one Login. Contact CUUG.">>badlogin
            mail $activename root<badlogin
            rm badlogin
        fi
    done
fi
```

Disk_pig

```
# performs a comparative analysis of previous disk usage vs current disk
# usage to find out who is being a "pig"

now=`date | cut -c5-7`
before=`ls -l pig2.new | cut -c33-35`
rightnow=`date | cut -c9-10`
earlier=`ls -l pig2.new | cut -c37-38`
withinweek=`expr $earlier + 7`
home_pct=`df | grep home | cut -c48-50`
if [ "$now" = "$before" ]
then
   if [ "$home_pct" -lt "91" -a "$rightnow" -lt "$withinweek" ]
   then
      exit
   fi
fi

sort -u -o temp2.new pig2.new
sort -u -o temp2.old pig2.old
rm newpigs.new
echo "`date | cut -c1-10`: Last months new overusers are : ">newpigs.new
comm -13 temp2.old temp2.new>>newpigs.new
rm -f temp2.old temp2.new pig2.old
cp pig2.new pig2.old
rm pig2.new
echo "">pig2.new
        echo "$user doesn`t exist in current directory." >> /home/coop/pig2.new
directories=`egrep '(100|150|200)' /etc/passwd | cut -f6 -d":"`
for user in $directories
do
   if [ -d "$user" ]
   then
      cd $user
      amt=`du -s | cut -f1 -d" "`
      if [ "$amt" -gt "1000" ]
      then
                  echo $user uses $amt
                  echo $user uses $amt >> /home/coop/pig2.new
      fi
   else
      echo " $user does not exist in the current directory structure"
   fi
done
```

Hogmail

```
# Send mail to all the users who are hogging disk space.

filen="pig2.new"
food="dskmes"
date >> hogmail.out
hogs=`cat $filen | cut -f3 -d"/" | cut -f1 -d" "`

for pig in $hogs
do
      if [ "$pig" != "byrnes" -a "$pig" != "empress" ]
        then
            mail $pig < $food
            echo $pig is a pig >> hogmail.out
      fi
done
```