The Roman Catacombs

BBS System for Linux

v.0.80

By Gregory Shaw
shaw@manwe.fmsoft.com

Table of Contents

Introduction 3
- Background 3
- Requirements 3
- Multiline capabilities 4
- External requirements 4
- Design decisions 4

Compilation and Configuration 4
- Pre-setup 4
  - Adding the BBS User and BBS Group 4
- Where to install? 5
- Unpacking the archive 5
- Compiling the System 6
- Adding the 'new' user logon 6
- Configuration 7
  - Linux FAQs 7
  - Modem configuration 7
  - Serial configuration 7
  - Initial Logon 8
  - Logon Scripts 9
  - System messages 10
  - Menus 10
  - Files Areas 14
  - CD-ROM Files Sections 16
  - Users 16
  - The Options (bbsinfo) File 16
  - Interfacing with External Programs 19
  - Uploading and Downloading 19

Testing and Normal Usage 20
- Using the System 20

Security 21
- How the BBS uses Unix security 21
- Security Holes - external programs 21

Being a System Operator (SysOp) 21
- SysOp Utilities 21
- Watching for Problems 23

Troubleshooting 24
- Potential Problems 24

Additional Information 24
- File Formats 24
- How to connect to FIDONet 26
- UUCP, Mail and News 26
- Other useful programs 26
- To Do 26
- More than two serial lines 26
- Multiple Machines and Large Setups 27
  - Monitoring through IP sockets 27
  - Terminal Servers 27
- The Roman Catacombs Setup 28

Updating from rocat 0.75 29

Conclusion 30

To Contact Me 30

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

Introduction

Welcome to the newest (working) BBS package available for Linux (Unix)! The Roman Catacombs BBS offers the following features:

- A familiar BBS interface for those who do not understand Unix.
- Batch upload and download through the use of external protocols.
- Automatic user addition to the BBS
- A very configurable BBS with many levels of security and access.
- Multiline capabilities
- Professional code quality with future maintenance in mind
- An elegant design that allows many additional features to be added without requiring a major rewrite of the code.

And many many more features.

Background

I've been running a BBS since late 1987. I started on Macintosh computers, and proceeded to write some externals for the BBS package that I was using. I wrote one game and a couple of utilities, most of which were shareware, and brought in enough money to keep the habit going.

In 1993, I got fed up with the Macintosh for two reasons:

1. The BBS was down if any sort of error occurred (e.g. a system crash)
2. The software was very limited and didn't do what I wanted to do with a BBS.

At that point, I was doing serious software development under Unix (Interactive SysVr3.2 for the curious). I was using C++, and thought that an object oriented design for a BBS would allow a lot of flexibility. So, I started developing a BBS.

I cut over to the BBS package late in 1993. It was a bit bare, and full of bugs, but it was a usable BBS system. I loved how any error in the BBS (e.g. a crash) logged out the user, and reset the line for the 'next' caller rather than crashing the operating system.

Late in the year, I got caught up in Linux. One of the major problems with writing the BBS for Interactive Unix was the number of users using the OS. I didn't want to create a BBS for a very small market. So, I cut over to Linux, and started the port.

After a few months of pulling my hair out because POSIX is different than BSD and SVID (and there are a few gaps in POSIX), I believe I've worked around most of the problems caused by POSIX.

That brings me to the present. The BBS is reliable, and I run a 3 line BBS on the software.

Of course, I don't claim that the BBS is bug free. FAR FROM IT! I do claim, however, that the BBS is a good solution, and really offers many options to a person wanting to setup a *reliable* BBS.

Requirements

The BBS has very few requirements, machine wise. Most of this has to do with Linux. If you can get Linux to run on your machine, and you've got a modem, you can generally use the rocat system. The basic requirements are:

386 or better
modem
development system installed (GNU c and c++) (for compiling from scratch)
5 megabytes of free disk space
familiar with basic Linux (Unix)
TCP/IP installed (minimally)
root privileges
man pages installed

Not required, but nice:

8 megabytes RAM or better (for compilation)

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

Multiline capabilities

Due to the inherent multi-user capabilities of Linux (Unix), it should be possible to have multiple persons using your BBS at the same time. rocat was designed with this in mind. It takes very little system resources to run rocat on your system.

External requirements

External programs are a major part of the BBS. Due to the flexibility available in editors, news readers and mail systems, all of this capability has been left out of the BBS, so that you can pick and choose your own 'flavor' of BBS. The externals that are necessary are:

editors (vi, emacs, pico ... pick your flavor)
mail readers (elm, mailx, mh ...)

Design decisions

I made some design decisions regarding this BBS. In other words, I wanted the BBS to be what I envisioned to be the 'perfect' BBS for myself. Some of the more esoteric features of some BBS packages haven't been built into rocat simply because I didn't feel they were necessary for my BBS. Some of the critical design decisions were:

- The Linux (Unix) system is used whenever possible. This is due to the sophistication of the system (and those features) as a whole. The following subsystems are not part of the BBS, but are expected to be used by the BBS:
  - Mail (sendmail, smail, elm, mailx, etc.)
  - News (fido, Internet news)
  - Chatting (talk, irc, etc.) (at this point)
  - Download protocols (rz, sz, kermit, xmodem, etc.)
  - Background processing (at, cron)
  - Login facilities (getty, login, etc.)
  - Modem control (flow control)
  - Passwords
- The BBS uses a modified 'system()' function. This assumes that the command executed is available and will behave properly. This has been done to allow interfacing with external system commands (see above). This is also a HUGE security hole, if used improperly.

Compilation and Configuration

Pre-setup

Adding the BBS User and BBS Group

A BBS administrator login and BBS administrator group is necessary for the BBS to be installed correctly. Do the following steps to add the BBS user and group:

1. su to root
   su
   < enter root password >
2. Execute the groupadd program in /usr/sbin
   /usr/sbin/groupadd bbs
3. Look at the file /etc/group. There is a number in the third field that is the GID (group id) for group BBS. Remember this number, you'll need it for adding the BBS Admin account.
   cat /etc/group
4. Execute the adduser program in /sbin.
   /sbin/adduser
5. The name of the user should be 'bbs'.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

6. The full name should be 'BBS Administrator', but can be anything you like (BBS Admin, etc.)
7. The GID should be the same as you saw in the /etc/group file above.
8. The UID (user id) can be anything, but I set mine up to start at 1000. Enter 1000.
9. The home directory should be /bbs/admin.
10. The shell can be your favorite shell (I prefer /bin/tcsh). You'll be using this account in the future, so use something that you're familiar with.
11. Enter a password you can remember. Please don't make it simple; crackers LOVE to eat BBSs for breakfast (at 3am).
12. In the unpacking step, the .cshrc and .profile for the BBS Administrator will be unpacked from the archive.

NOTE: All further steps (unless otherwise noted) should be done by the BBS Administrator account. He will 'own' all of the files involved with the BBS, so you should get used to being 'him' (or 'her'). You may get to that user by the following command:

```
su - bbs
< enter BBS Administrator password >
```

Or, alternatively, you may login as 'bbs'.

(sub-note: don't do it until the BBS Administrator's account is available. It won't be available until the next step is completed)

Where to install?

Installation of the BBS should be done on a separate disk partition, if possible. This allows the BBS to exist in it's own 'area'. The following is the setup of The Roman Catacombs BBS (my BBS):

```
/       boot disk
/usr    another disk
/bbs    the BBS disk.
```

The BBS disk contains everything relating to the BBS, executables, headers, files, etc.

I'd recommend using a separate partition, if at all possible. This will allow the BBS to use the disk caching of the device.

If that isn't possible, pick a place to install the bbs, and make a symbolic link to /bbs.

This is accomplished by the following steps:

1. mkdir my_bbs_dir
2. cd /
3. su (you need root privileges to make a link in the root directory)
4. ln -s path_to_my_bbs_dir/my_bbs_dir /bbs

You'll also want to make sure that the BBS Administrator and the BBS group own the BBS home directory:

```
chown bbs.bbs path_to_my_bbs_dir/my_bbs_dir
```

Note: At this point, you'll want to make the directory /bbs/admin. This is where the BBS Administrator's home directory is found. You won't be able to login as 'bbs' without a home directory. It should also be owned by the BBS Administrator and group.

Unpacking the archive

The archive will come in a file named 'rocat.tar.gz' or 'rocat.tgz'. Both of these filenames represent the same file compression. To uncompress:

1. su to the BBS Administrator (all files should be owned by the administrator):

   ```
   su bbs
   <enter the BBS Administrator password>
   ```

2. Copy the file to the BBS directory.

   ```
   cp /some_path/rocat.tar.gz /bbs
   ```

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

3. Change directory to /bbs
    cd /bbs
4. Untar the BBS.
    tar xvfoz rocat.tar.gz
    Note: If tar complains about 'unknown option z', you need to gunzip the file prior to untarring the file:
    gunzip rocat.tar

At this point, much of the BBS structure will be created. Additionally, the BBS Administrator's setup information (.cshrc, .profile, etc) will be copied to /bbs/admin. You'll want to log out and log back in as the BBS Administrator before continuing the process.

Compiling the System

Compiling the system is the easy part. There is very little that needs to be done prior to compiling the code. There are some limits setup in the main header file and spread out here and there in the code. For the most part, you won't need to mess with these settings. If you really feel you need to muck around with the code, feel free. Be warned, however, that the rocat system is a complex beast, and a very fickle one, if changed indiscriminately. Save an original copy of the BBS prior to making any changes.

The following steps should result in a set of BBS executables:

1. CD to the src directory.
    cd src/rocat
2. Do a make depend.
    make depend
3. Do a make. This should start the build process.
    make

Come back in a couple minutes (on a 486/66 with 20 megabytes of memory) or a couple of hours (386SX-16 with 2 megabytes of memory). If you get errors, make sure of the following:

1. The development system is installed. This includes GNU C, GNU C++ and GNU LIBG++. rocat is written in C++, and requires all of the C++ facilities to compile.
2. Your path has the compilers in it. 'make' depends on your path to execute commands. If you can't execute the command, make won't be able to execute it either. Your path should contain (at least) /bin, /usr/bin, and /usr/local/bin.
3. bbs owns the entire rocat directory structure. If bbs doesn't own the directories, you won't be able to make any changes to the files and/or directories. (and generally, things will be pretty broken from here on)

Adding the 'new' user logon

The 'new' user logon must be added to allow users to log into your BBS for the first time. To add the 'new' user logon, do the following steps.

1. su to root
    su
    < enter root password >
2. Look at the file /etc/group. There is a number in the third field that is the GID (group id) for group BBS. Remember this number, you'll need it for adding the 'new' account.
    cat /etc/group
3. Execute the adduser program in /sbin.
    /sbin/adduser
4. The name of the user should be 'new'.
5. The full name should be 'New BBS User', but can be anything you like (New User, etc)
6. The GID should be the same as you saw in the /etc/group file above.
7. The UID (user id) can be anything. You'll be changing the UID, so enter something that isn't already in use. Say 10000.

8. The home directory should be /bbs.
9. The shell should be /bbs/scripts/login.scr. This is important.
10. Enter a simple password to setup the account.
11. Now comes the tricks. cd to /etc
12. Edit the passwd file.
    vi passwd
13. search for the 'new' user.
    /new
14. You need to change the 10000 in the 'new' user's ID line to 0. The third field is the number to change (the file is delimited with colons).
15. Delete the password found in the second field. The final entry should look like this:

```
new::0:104:New User Account:/bbs:/bbs/scripts/login.scr
```

The new user is now installed. However, if you have shadow passwords installed, you'll need to edit the file /etc/shadow to delete the password for new.

If you do not delete the 'new' user's password, nobody will be able to log in as 'new', because they won't know the password. If there is no password, it skips the password prompt.

Configuration

Linux FAQs

Note: In the below configuration and examples, I'm assuming that you're familiar with adding modems and configuring gettys.

If you're not familiar with these items, I recommend the Linux FAQs. They're invaluable information for those starting from scratch. The FAQs are available as part of the Slackware distribution, and, should you install them, are found in /usr/doc/faq. If you don't use Slackware, and have access to news, please look in the comp.news.answers newsgroup for Linux FAQs.

Modem configuration

Configuration of the modems is very important to the BBS. If the modem isn't configured correctly, the BBS will never know that there is a user on the system!

At this point, I'm using a ZyXEL U-1496+ on my BBS. However, I won't go into the configuration of the modem in detail; all you really need to know is the generic parameters that are necessary for the modem to work. I've included the items that are generic to most modems in parenthesis at the end of the description. The following are the key configuration items for your modem:

A GOOD MODEM CABLE (one that has all pins straight through is the best)
Serial speed (DTE-DCE) speed should be locked at 19.2Kbps or faster speed.
Hardware flow control should be on.
The modem should reset upon loss of DTR. (AT&D3)
The modem should respond to carrier-detect (CD). (AT&C1)
The modem should answer the phone on the first ring. (ATS0=1)
If possible, error-free and data compression should be enabled.

It's possible (and sadly, probable) that some of the above items will go by different names.

Remember the speed you've locked the modem at. You'll need it for the serial configuration.

Serial configuration

Serial configuration involves telling the computer where your modems are, and, what speed they're running at. At this point, I'm assuming you've got your modem(s) configured, and are ready to test the logon capabilities of your machine.

Serial configuration involves the following steps:

1. Configuring the /etc/gettydefs file.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

2. Turning on 'getty' processes for your modem.
3. Changing the /etc/issue file
4. Testing (if possible)

Configuration of the /etc/gettydefs file is easy. Simply add the following line to your gettydefs file:

```
#  Modem locked at 38400:
#
38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP CRTSCTS #@S login:
#38400
```

Of course, if you are running at a different baud rate, you'll want to change all occurrences in the above file to the baud rate you wish to use.

To turn on a 'getty' process for your modem, edit the /etc/inittab file. In that file you should see something similar:

```
# Serial lines
s1:45:respawn:/etc/uugetty -t 90 ttyS0 38400
s2:45:respawn:/etc/uugetty -t 90 ttyS1 38400
```

The above two lines are for COM1 and COM2 ports (DOS-style). The above lines mean:

1. Spawn a 'getty' for the port ttyS0 running from the '38400' gettytab entry when in run level 4 or 5.
2. If you get a carrier detect on the line, and nothing happens for 90 seconds, go ahead and hang up the phone.

Note: The system goes into mode 4 or 5 by default on most systems.

Note: Make sure that the files /dev/ttyS0 and /dev/ttyS1 exist. They should look like this:

```
crw-rw-rw-   1 root     root       4,  64 May 21 19:43 /dev/ttyS0
crw-rw-rw-   1 root     root       4,  65 May 21 19:43 /dev/ttyS1
```

If these files don't exist, the 'getty' program will error off, and you'll see errors on the console regarding the problem entry. If they don't exist, see the Linux FAQs for creating device files.

If you make any changes in the /etc/inittab file, be sure to issue a 'init q' command so that init will re-read the file and recognize the changes.

Note: For new users to log on (via the 'new' account), the tty line must be entered in the /etc/securetty file. The 'new' user runs as root, and without the securetty entry, root logon will be denied on that tty. Do a 'man login' for more information.

A default issue file can be found in /bbs/text/issue. Edit this file and copy it to /etc/issue to activate the message prior to logon.

If possible, you should have a friend log into your BBS. If you have trouble, please reference the FAQs about turning on dialup services.

Initial Logon

The initial logon process is documented below:

1. The user logs in as 'new'.
2. The user is given a 'welcome to the BBS' message.
3. He is asked if he already has an account, and is unable to logon.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

4. If this is the case, he's asked for some information, and is able to send mail to the SysOp of the BBS. He is then logged out.

5. If the above was not the case, he's shown a file (/bbs/text/bbswelcome) describing the focus of the BBS.

6. He is asked "Would you like to become part of the BBS?".

7. If anything other than 'Yes' (or something starting with 'y') is entered, he is logged out.

8. He is then asked his first name and last name. Checking is done from the file /bbs/text/badwords to make sure that there isn't an illegal word in his name. This prevents logins of the form 'Doctor Death' and 'Joe Mama' from becoming usable logins.

9. His first name and last name are place together to form a unique login ID for him. It generally takes the form first inital and last name, but if you should have a logon of that nature, it will attempt to put additional first name characters along with the last name. Note that only 8 characters are used. A sample:

Name	Logon
John Smith	jsmith
John Smith (number two)	josmith
Alfred Neumann	aneumann
Alfred Neumann (two)	alneuman

Note: It is possible to get in a loop. If you have a short name, and all possible combinations have been used, it will cycle endlessly, trying for a combination. Since this is very improbable, I've left this 'bug' in the system.

10. The user is given his login ID.

11. The user is prompted for a password. It will loop until a valid password (as defined by the password program) is entered.

12. The user account is added (via the /bbs/scripts/addbbsuser script).

13. The user is given a "please log in again with your new account" message and logged off.

When he logs back in with his new account, the following happens:

1. Since he is not found in the userlog (/bbs/admin/userlog), the BBS asks whether he wants to be part of the BBS.

2. If not, he's logged off.

3. He is asked for the city he is calling from.

4. He is asked for the state he is calling from.

5. He is shown a message describing what terminal types are, and asked for the terminal type his software supports.

6. He is asked whether his software supports color. (this isn't used at this time)

7. He is shown a file describing the editors available on the system (/bbs/text/editors) and asked which one he'd like to use.

8. He is asked for an alias.

9. He is asked how many lines are available on his screen.

10. He is asked how many columns are available on his screen.

11. He is prompted with the above information, and asked whether it is correct. If not, it goes back to 3.

12. He is logged into the BBS (saved).

13. He is shown the 'welcome' message. (/bbs/text/welcome)

14. His personal information is displayed.

15. He is walked through system1.msg through system3.msg.
    (Note: this can be turned off through the config file)

15. He is given an optional 'fortune'.

16. At this point, the BBS runs the menu 'main', and the BBS operates normally.

Logon Scripts

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

You'll want to edit your logon script to represent 'your' BBS. The script I am talking about is found in /bbs/scripts/login.scr. There is a 'Welcome to The Roman Catacombs!' message found in there that you'll want to change to your BBS name.

You'll also want to edit the '/bbs/bbs' file. This file is the pipe that allows users to use the BBS effectively. The file is as follows:

```
# file to execute prior to starting main bbs program
#


IFS=""
PATH=/bin:/usr/bin:/usr/games:/usr/local/bin:/usr/ucb
export PATH
BBSDIR=/bbs
SYSOP=shaw
VISUAL=pico
EDITOR=pico
export BBSDIR SYSOP VISUAL EDITOR IFS
SHELL=$BBSDIR/rocat
export SHELL

# hack to get around the getty's problem with not setting CRTSCTS
# setup default terminal settings

stty erase "^H" kill "^U" intr "^C" eof "^D"
stty hupcl ixon ixoff crtscts

exec $SHELL
```

In the above file, you should change the SYSOP variable, and any other options you feel it is necessary to change. I've got the editor set to 'pico' at this point, as it is an easy to use editor with excellent help.

System messages

Note: All text files are found in /bbs/text.

There are quite a few text files that are used regularly in the BBS. They are:

Filename	Contents
badwords	those words deemed inappropriate for logon names
bbsinfo	a short description of the BBS hardware (optional)
bbswelcome	a short blurb on the BBS used prior to logging in the first time
editors	a list of editors available on the system
logoff	the logoff message given to the user upon exit
newuser.msg	shown to the user on initial logon
system1-3.msg	status messages about the BBS (optional)
welcome	a mood-setting intro blurb shown on each logon

All of the above files are customizable in any fashion you see fit.

You'll end up modifying almost all of them.

Menus

The menus are how a user interacts with the BBS program. Because of this, the entire 'feel' of your BBS is how you setup the menus. This is the place where you differentiate your BBS from other BBSs. Well, enough soap-box. Let's get started.

When the BBS program starts, it displays the user information, the system messages, and then calls the mandatory 'main' menu. This menu serves as the keystone menu; a place to start.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

Let me start with what a menu does. Basically, a BBS menu shows the user a list of options, and, when an option is selected, tells the BBS to do something. That something can be as simple as 'transfer to another menu', or as complex as 'call the external mailer program to send a message to the SysOp'. It is all up to you on how complex you wish your menus to be.

A menu, as described above, is made up of 'options'. These options take the form of individual items, each found on an individual line. A typical menu looks like the following:

```
0|||||| Roman Catacombs
Entry Vault
0|||||| ------------------------------------------------------------
------------
0||||||
0||||||                                      Exit
0||||||
0||||||                                         <G>
0||||||                                ----------------------------
0||||||                                H>elp
0||||||                           <W> P>rivate Mail              <E>
0||||||             Art Studio -- R>egistration Instructions -- Computer
Files
0||||||                                U>tilities
0| 400 >||||                             S<Y>sop's Area
0||||||                                C>hat with SysOp
0||||||                                ----------------------------
0||||||                                         <S>
0||||||
0||||||                                    News and Games
0||||||
0|||||| ------------------------------------------------------------
------------
0|||||| <-> Previous Menu     <F> Feedback to SysOp   <G> Quick Logout
1|g|||||
3|-|||||
8|c|||||
2|y|400|>|||sysop|
2|e||||computerfiles|
2|w||||art_studio|
2|s||||newsngames|
2|u||||utilities|
21|f||||elm -s feedback shaw|
2|p||||mail|
10|r||||reg.instr|
2|h||||helptop|
2|h|||||helptop|
```

The above is the main menu for The Roman Catacombs.

Let's dissect a particular line:

```
0|||||| Roman Catacombs
Entry Vault
```

In the above line, there is a number, followed by a number of pipe-letters (|), followed by some text. In the above line, you find eight fields separated by the pipe letters. The fields are:

1. Command Number
2. Key used to select item
3. Access level to see item
4. Access level modifier
5. Flags required to see item
6. Flags modifier
7. Optional data
8. The text for the item.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

In the above line, 0 is the command number (which is a text-only command), and fields 2-7 are blank (e.g. unused), and field 8 is the text to display.

The user doesn't see anything but field 8. If field 8 is empty, he sees nothing.

When a field is blank, it's assumed that everything is available. For example, field 2 was the key to select the item. Since it's an output only item, having a key there would be pretty useless, hence it's empty.

Let me define the fields in human terms:

Field	Formal Description	Use
1.	Command Number	Tell the BBS what operation to perform
2.	Selection Key	Key used to select menu item.
3.	Access Level	Access level required to see/use item.
4.	Access Flags	Define how to use Access Level
5.	Flags	Flags necessary to see/use item.
6.	Flags Modifier	Define how to use Flags.
7.	Optional Data	(command specific usage)
8.	Text	Text to show user.

Let me do a few more examples:

```
2|e|||||computerfiles|
```

In the above line, 2 is the command (which means change to another menu), 'e' is the key used to activate the item, and all the rest of the fields are null. Note how the 'e' goes with the item found above it:

```
0|||||||                    <W> P>rivate Mail              <E>
0|||||||          Art Studio -- R>egistration Instructions -- Computer
Files
```

The user would select 'e' to go to the 'Computer Files' area (a menu in this case).

This is how menus work. They define the text that a user sees, and, define what happens when a user activates that menu item.

Of course, there are times when you want certain users to be able to use a command, and, times when only you should be able to use a command. Generally, these are defined by the security of the BBS. There are commands that should only be executed only by the SysOp (or someone very trustworthy). For this reason, there are fields 3 through 6. These fields allow you to define who gets to see and use what commands, based on access level and flags. Let me define fields 3 through 6 a little better:

Field 3 sets the access level for the command. This and field 4 define the access level required to see/use the command. An example:

```
2|y|400|>|||sysop|
```

The above item defines a switch to another menu, the 'SysOp' menu. Of course, you don't want just anybody accessing this menu. That is why the 400 and the '>' symbol are there. The 400 is the access level (ACL), and the '>' means that the user's access level must be greater than or equal to 400 to see and access this menu item.

By the same token, the flags required to see/use an item are selected the same way:

```
2|y|||1|=|sysop|
```

The above item has flag 1 in the flags area and '=' in the modifiers. This means that the user must have flag 1 set before he may see/use the item.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

The fields will accept the following:

Field	Valid Contents	Meaning
ACL	0-32767	
ACL Modifier	>	User must have ACL greater or equal to ACL
	<	User must have ACL less than ACL
	=	User must have ACL equal to ACL
Flags	1-32	
Flags Modifier	=	User must have flag set
	!	User must not have flag set

Based on the above, you can have all sorts of permutations based on the user's capabilities.

The official definitions for the command numbers (the first field) follows:

Command	Description	Optional Data
0	Output only	n/a
1	Log out of BBS	n/a
2	Branch to another menu	menu name
3	Exit to previous menu	n/a
4	Search userlog for user information	n/a
5	List users in userlog	n/a
6	Display user information	n/a
7	Re-enter user information	n/a
8	Chat with SysOp	n/a
9	Not Used	
10	Display a file (paged)	filename
11	Display a file (not paged)	filename
12-20	Not Used	
21	Launch external program	external program
22-24	Not Used	
25	List new files with option to download	files section
26	List new files without option to download	files section
27	List all files with option to download	files section
28	List all files without option to download	files section
29	Search for string in files area (w/ option)	files section
30	" " (w/o option)	files section
31	View detailed information on file	files section
32	Download file(s)	files section
33	Upload file(s)	files section
34	Delete a file that you uploaded (not implemented)	
35	Download a single file (with ratio checking)	full path and filename
36	Download a single file (without ratio checking)	full patch and filename
36-39	Not Used	
40	Search for users to edit/delete	n/a
41	Delete inactive users	n/a
42	List inactive users	n/a
43	List users for edit/delete	n/a

Now that you've seen the entire list (the short form), you know what is available for your menus. Let's go on to dissect the rest of the main menu:

```
0|||||||                          H>elp
```

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

```
0|||||||                      <W> P>rivate Mail              <E>
0|||||||          Art Studio -- R>egistration Instructions -- Computer
Files
0||||||| ------------------------------------------------------------
------------
0|||||| <-> Previous Menu     <F> Feedback to SysOp   <G> Quick Logout
1|g||||||
3|-||||||
8|c||||||
2|e|||||computerfiles|
2|w|||||art_studio|
21|f||||elm -s feedback shaw|
2|p|||||mail|
10|r||||reg.instr|
2|h|||||helptop|
```

In the above menu fragment , you see a number of menu items. Let me first say that I've taken the philosophy of breaking the text display and the commands into two different sections. This allows me to see what the user sees without going into the BBS to check. The first 3 items are the 'text' that is shown to the user to select an item. The last 9 items are the items that are available for selection.

Help is a menu transfer command (2) and will transfer the user to the menu 'helptop'.

Hitting 'w' will move the user to the 'art studio' (a place for picture files). The name of the menu is 'art_studio'. This name is the same name found in the directory /bbs/menus. It is assumed that all menus are found in the /bbs/menus directory.

A completely different command is command 8, which will execute the 'chat' program with the sysop's name. At this time, I have it configured to use 'talk', but you may change this as you wish. See the 'config' file section on how to do this.

Specifically note command 21, which executes feedback to the SysOp. This command is simply a 'execute external' command. This command is passed to the system as if it were typed from and command line. In this case, it is doing 'elm -s feedback shaw', which calls the 'elm' mailer (an easy to use mailer) with 'feedback' as the subject of a message to the SysOp (shaw).

A special note about menu entry-exit. The proper way to enter/exit menus is to use a command '2' to transfer to a menu, and a command '3' to return from that menu. This is so that if you have many menus going to the same location, when you exit that menu, you return to where you were last. In computer terms, the menu system is a 'stack' of menus. When transferring to a new menu, that menu is put on 'top' of the stack. When you exit that menu, the old menu is taken from the stack, returning you to the last previous menu. Note that it is possible to transfer between menus using '2' commands all the time; however, it is *not* a good idea. The stack will continue to fill, and when it hits 50 (the current stack size), the program will probably die.

A small sub-note. It is not necessary for the stack of menus to be empty when the user logs out of the BBS. They can log out at any time without causing problems with the BBS. (In fact, they can drop carrier at any time without causing problems)

Files Areas

A major part of most BBS systems is the ability to send and receive files. Of course, this functionality is built into the rocat BBS system. The BBS supports all external protocols that are possible on Linux (Unix) systems.

Configuration of the files system involves two steps:

1. Configuration of the external protocols file. (/bbs/config/protocols)
2. Configuration of the main BBS files header file (/bbs/filehdr/bbs_files_hdr)

A sample protocols files follows:

```
# this file describes the external protocols to be used by the bbs and
```

```
# the command to be used to invoke it.
# the format is:
# UPLOAD|command|needs filename (y or n)|key to select|text description
for user
# DOWNLOAD|command|key to select|text description for user
# Note that if a command needs a filename, multiple file uploads are not
# allowed.  (this is the case with the below protocols.  If something
# different is found, then it can be changed)

U|rz|n|1|1. Zmodem
U|rb|n|2|2. Ymodem
U|rx|y|3|3. Xmodem
U|kermit -r|y|4|4. Kermit

D|sz|1|1. Zmodem
D|sb|2|2. Ymodem
D|sb -k|3|3. Ymodem-1K
D|sb|4|4. Ymodem-G
D|sx|5|5. Xmodem
D|sx -k|6|6. Xmodem-1K
D|kermit -s|7|7. Kermit
```

As you see above, to add additional protocols, simply enter the upload and download commands as described in the above file.

Configuration of the main BBS files header file is also simple. Another sample file follows:

```
#
# files header for the BBS.  Should contain one entry for every files
area
# on the BBS
#
# format: filename [R|C] absolute_path_to_files.bbs sysop_name
access_level down load_path upload_path age_to_delete_files
long_name_of_area
#
# dashes fill non applicable positions

# dos files areas
[ dosapps R - shaw 0 dos/apps uploads 0 Dos Applications ]
[ doscomm R - shaw 0 dos/comm uploads 0 Dos Communications Programs ]
[ doscomp R - shaw 0 dos/compress uploads 0 Dos Compression Programs ]
[ dosdrivrs R - shaw 0 dos/drivrs uploads 0 Dos Drivers ]
[ cdromgames C /cdrom/dos/games/files.bbs shaw 0 dos/games uploads 0 Dos Games ]
[ dossounds R - shaw 0 dos/sounds uploads 0 Dos Sounds ]
[ dosutils R - shaw 0 dos/utils uploads 0 Dos Utilities ]
```

This one is a bit more tricky. The fields that make up the file are:

1. The name of the files section. This is the name entered in field 7 of a menu. This is also the name of the associated files header found in /bbs/filehdr when the section is a rocat section.
2. The type of files section C for CDROM or R for rocat.
3. The absolute path to the files.bbs file for CDROM sections.
4. The sysop's name. Not used at this point.
5. The access level required to use the files section. Note that this is a minimal number; if a user's access level is below this value, he will be denied access.
6. The relative path to the directory where the files are stored. Note that you must create this directory yourself. The whole path is /bbs/files/relative_path.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

ex: dosapps is found at /bbs/files/dos/apps.

7. Age to automatically delete files found in the section. This option is available for material that is dated (such as ski reports) and that can be automatically deleted after some period (in days). Note that this operation isn't supported as yet.

8. Name of the files section (as reported to users). This can be a very long name, but it would be better if it were short. Most programs limit it to about 20 characters.

CD-ROM Files Sections

As you can see above, the rocat BBS system supports CD-ROM (read-only) files sections. These sections are unique in that they never change (until you get a new CD-ROM) and they already have filenames and descriptions on the CD-ROM.

rocat will read the files.bbs that is found in every directory that has BBS readible files. Simply enter the files.bbs as shown in the above example. Note that a files section header file is not required, nor is it generated.

There are some extenuating circumstances that rocat checks for automatically:

Should the CD-ROM be unavailable (you forgot to mount it or you have another CD-ROM online in it's place), access of the CD-ROM section will generate a message to the user telling him that the section is unavailable.

Uploads are not possible for CD-ROM sections. I have made the assumption that you have your own files sections that would be used for uploading files that would reside in that section.

Other than that, CD-ROM filesections operate as they would with a rocat files section.

Users

The users of a BBS must have some place to store their configuration files and any other miscellaneous files. This is accomplished through the use of user directories. The current implementation uses the following directory structure:

/bbs/users - the root directory for BBS users.

[a-z] - the first initial of the user's name.

An example: If Alfred E. Neumann logs onto the BBS through the 'new' user logon, his home directory would be:

/bbs/users/a/aneumann

The initial logon scripts copy the following files to the user's account:

/etc/stdprofile as $HOME/.profile
/etc/stdlogin as $HOME/.login
/etc/stdcshrc as $HOME/.cshrc

Note that these files aren't used when the user has the BBS as his shell. These files will only be used should the user get a shell (csh, bash, tcsh, etc).

Note also that these files don't exist as a standard Linux item. You'll need to configure these scripts when you figure out how you want to handle shell-logins. (e.g. users that log into the Linux (Unix) OS rather than the BBS)

The Options (bbsinfo) File

Almost all options for the BBS program are found in the options file, /bbs/config/bbsinfo. The file is a simple text file, with comments describing what the features are, and how to enter the features. The features file found on the Roman Catacombs BBS follows:

```
# this file contains the bbs pathing and machine information
```

```
# setup as you desire
# it should be in the format: NAME <tab> value
# blanks and lines starting with # are ignored

# host where error logger is running

LOGHOST manwe

# host where sysop's watch program is running

WATCHHOST       manwe

# upload to download ratio      (7.0 is 7 downloads for each upload)
# negative numbers refers to how the ratio is done - if it's
# negative, then the ratio is enforced *BEFORE* the user downloads
# the ratio number of files (in the case of -7.0, he has to upload
# before he can download the 7 files possible.
# normal (aka positive) numbers refers to a normal upload ratio.
# meaning the ratio won't be enforced until after the even number
# of files has been downloaded (e.g. on the 8th file he wouldn't
# be able to download.
# set below to 1000 or some such number for no ratio.
RATIO   7.0

# default access level for a new user on the bbs

DEFACL  100

# default terminal type for new user on BBS

DEFTERM ansi

# time that a user's time limit is good for.
# this is the amount of time that a user may use his 60 minutes for.
# ex: if you set it to 24 hours, he gets 60 minutes every 24 hours.

WAITTIME        12

# credit chat time with sysop?
# 0 for false, 1 for true

CREDITCHAT      1

# credit upload time?
CREDITUPLOADS   1

# the pager to use on your system (use 'more' or 'less' or whatever
# you like).  Note that the default user path must have this command
# available for it to be accessible.  (e.g. if you have 'less' in
# /usr/local/bin, /usr/local/bin *MUST* be part of the default login
path
# Note:  Fully qualifying the path won't work.  Only 15 chars are
allocated
# to the 'pager' variable.

SYSTEMPAGER     more

# login name of the sysop

SYSOP   shaw

# the following flags determine whether to show the user the system.X
# (where X < 4) messages are shown to the user at each logon or only
once
```

```
# if 1, message will be shown at initial logon only.  0 for every time

SYSTEM1MSGONCE  0
SYSTEM2MSGONCE  0
SYSTEM3MSGONCE  0
# sysop's chat hours.  If the user hits 'chat' outside of these hours,
he
# will be told you are not available and to leave a message.
# time is in military.  1900 is 7pm, 2200 is 10pm

CHATON  0000
CHATOFF 2400

# the below are the different things you can put on the command prompt
# line when at the end of a menu.
# turn on SHOWTIMELEFT to show the amount of time the user has left
# turn on SHOWVALIDKEYS to show the valid hot keys to the user
# USERPROMPT is the text that will prompt the user to hit a key.
# ex: with all turned on,
#  (15 Minutes left) Command? (a,b,d,i,k,t)  <--- user prompt here

SHOWTIMELEFT    1
SHOWVALIDKEYS   1
USERPROMPT      Command?

# the following determines the default card type for a new user
DEFCARD blue
# the following determine the amount of time and downloads that a user
# may have for different access levels.  Define as you wish.
# the format is this:
# card color, access level, timelimit, additional flags, amount of
downloads per day
# flags is a hex value that will be added to the user's flags upon logon
# (e.g. for flag 12 on for additional access, it would be 400)
# a -1 value in a limit field means 'don't enforce'
REDCARD         100     20      0       0
BLUECARD        200     40      0       500
GREENCARD       200     40      1       500
WHITECARD       400     60      1       -1
# I don't use the below.  Use them as you want.
GREYCARD        500     0       0       0
PINKCARD        600     0       0       0
YELLOWCARD      700     0       0       0
# this is the sysop's card
BLACKCARD       10000   255     0       -1

# location of the mail spool file
MAILSPOOL       /usr/spool/mail
# minimum wait time between new mail checks (in seconds)
MAILCHECK       30

# 'talk' program for chatting with the SysOp
TALKPROG        talk

# show the user his 'fortune' upon logon?
SHOWFORTUNE     1

# Number of minutes of inactivity before user logged out
# 0 for no inactivity
INACTIVITY      5

# Max number of K the user may download with one batch
MAXK    2500
```

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

```
# don't delete this line!
```

As you can see, there are quite a few options. Edit the above file however you wish. Note that at this time, there are only 8 user card colors. No more colors can be added without changing the code.

Note: If you want to edit the user to give him additional access (say access 500 when he's using a blue card with access 200), the BBS will *not* drop his access. The cards are used as a minimal access value system only. If a value is less than a card's value, it will be set to the card's value. If a value is higher than a card's value, it will be left alone. This is to assume that you (the SysOp) have changed his access level for some reason. This is true for all values found in the cards.

Interfacing with External Programs

Interfacing with external programs is simple. When a user selects an external program, that program is called from within the BBS. To him, it appears like the program is built into the BBS. See command (21) above for more information.

WARNING: External programs that are not compiled with security turned on are **huge** security holes. An example that I found recently:

I use the 'tin' news reader. As part of tin, it's possible to do a shell command (via the '!' character). If you compile tin to turn this off (which you should), then you will not have problems. However, I was able to type 'chsh -s /bin/tcsh', and the shell command changed my shell from the BBS to a shell account. It was this simple.

Solution (in this case): I turned off 'chsh'. I am looking into 'rsh' (restricted shell) at this point. It doesn't appear that Linux supports 'rsh' today, but I am still researching this.

WARNING: When you put an external command in your BBS, you *MUST* test this command for shell-giving options. If you do not, you are opening your system to crackers; people who break systems for the fun of it.

Uploading and Downloading

Uploading and download to/from the BBS is meant to be as transparent as possible. (see the files areas section for configuration of files areas) A typical uploading session follows:

The user lists files in a files section.

He sees a number of files he's interested in. He 'Marks' the files for download.

He then hit's 'Download'. He is given a list of the files he has marked.

He then is given the list of protocols on the system. He selects the protocol he wishes to use.

The download commences. Note: at this time, all downloads are assumed to be successful. This means that if the user downloads 5 files, yet only gets 1 downloaded successfully, he is marked for 5 downloads. This error is due to external file transfer programs not returning any useful information as to how many files were actually sent.

The user's information is updated accordingly to reflect the files (and size of the files) downloaded.

Of course, the above is a simplistic view of how to download. Behind the scenes, there are a lot of security features that must be satisfied prior to a download taking place:

1. The user must be able to download (via the menu option).
2. He must have enough K available (via his card color) to download the file(s).
3. He must keep his upload/download RATIO within BBS specifications.
4. He must not have more than MAXK of files selected for download.
5. He has the access level to access the files area (see files areas).

Note that uploads are credited to the user's account (provided that option is selected via the options file). An example:

The user has a blue card. This enables him to 40 minutes on-line, and 500K of download space. The user decides to download the following files:

Filename	Size
a_big_gif.gif	450K
another_big_gif	75K

He would be denied the second file, because that would be over 500K of transfers. Note that if he downloaded the 450K file and waited the expiration time (12 hours in the example options file), he would be able to download the other file.

Uploads are handled a bit differently. Of course, a user can upload as many files as he wishes. When the user wants to upload, he goes to the section he wishes to place the files, and selects the 'Upload' option. The below steps then happen:

1. He is then prompted for a file transfer protocol.
2. After he selects the protocols, a new directory with the user's name is created in /bbs/tmp.
3. The protocol is then called with the upload option.
4. After the exit of the transfer program, the user is asked whether the upload was successful.
5. If the upload was not successful, the user is asked whether to keep the partially uploaded files.
6. If the upload was successful, the user is prompted for each uploaded file's description.
7. As the file descriptions are entered, the files are moved to the uploads directory (as defined in /bbs/filehdr/bbs_files_hdr for the section).
8. If the 'credit uploads' option is turned on in the configuration file, the user is credited for upload space and upload time.

Testing and Normal Usage

Using the System

Generally, a user will log into the system and use it normally without problems. However, there exceptions. When a user has a shell account, and wishes to access the BBS, how does he do that? How does he upload without uploading through the BBS itself?

The answer to these questions is to mimic how the BBS works normally.

For a user to log into the BBS from a shell account, he can type

```
/bbs/bbs
```

And he will see the BBS just like everybody that has /bbs/bbs as their login shell.

Similarly, uploading to the BBS via the command line is accomplished by doing what the BBS does normally. When a user uploads via the command line, he should do the following:

1. Upload the files as he normally would via the command line.
   ```
   rz
   ```
2. Make a directory in /bbs/tmp with his login name as the name.
   ```
   mkdir /bbs/tmp/myloginname
   ```
3. Copy the files that are for a particular files section to that directory.
   ```
   cp * /bbs/tmp/myloginname
   ```
4. Enter the BBS.
   ```
   /bbs/bbs
   ```
5. Go to the files area where the files are to be put.
6. Select 'upload'.
7. Select zmodem as the protocol.
8. Type ctrl-x a number of times. This will abort the download.
9. The BBS will ask 'Was the upload successful?'. Type 'y'.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

10. The BBS will then prompt you for descriptions for all files found in the /bbs/tmp/myloginname directory.

Security

How the BBS uses Unix security

The BBS works with the Unix security system so that unusual protection methods are not necessary. Basically, the BBS allows users to access and change BBS files via running set-group-id (SGID) 'bbs'. All files that can be modified by users within the BBS must be owned by group BBS, and writable by group BBS. Directories must have the group write and group execute permissions for user access. The directory structure and permissions follow:

The BBS root directory:

```
drwxr-xr-x   7 bbs      bbs          1024 May 22 22:17 admin/
-rwxr-xr-x   1 bbs      users         516 May 16 20:32 bbs*
drwxr-xr-x   2 bbs      bbs          1024 May 22 22:01 bin/
drwxr-xr-x   2 bbs      bbs          1024 May 25 17:21 config/
drwxrwxr-x   2 bbs      bbs          2048 May 22 22:10 filehdr/
drwxr-xr-x   8 bbs      bbs          1024 Apr 17 11:54 files/
drwxr-xr-x   3 bbs      bbs          2048 May 17 21:01 menus/
drwxr-xr-x   2 bbs      bbs          1024 Feb 23 17:40 new/
drwxr-xr-x   5 bbs      bbs          1024 Feb 23 16:55 othermounts/
-rwxr-sr-x   1 bbs      bbs        259076 May 22 21:32 rocat*
drwxr-xr-x   2 bbs      bbs          1024 May 21 22:04 scripts/
drwxr-xr-x  17 root     root         1024 May 15 20:00 spool/
drwxr-xr-x   2 bbs      bbs          1024 Apr 20 22:10 text/
drwxrwxrwt   3 bbs      bbs          1024 May 20 01:08 tmp/
drwxrwxr-x   2 bbs      bbs          1024 May 22 22:07 uploads/
drwxr-xr-x  28 bbs      bbs          1024 Feb 24 20:09 users/
```

Note that rocat (the BBS program itself) is SGID. Everything else that needs to be writable (filehdr for files headers, and uploads for new uploads) are writable by the BBS group. Nothing else should be writable by the BBS group, nor should it require BBS write access.

Security Holes - external programs

I can't stress this enough: WATCH OUT FOR EXTERNAL PROGRAMS. This is how many crackers gain unauthorized access to your system.

Being a System Operator (SysOp)

SysOp Utilities

There are a number of utilities I've written to make the BBS easier to maintain. They are:

Program	Description
fileutil	a utility for maintaining files areas
errlogd	the BBS error logger
monitor	a utility for watching users using the BBS

Let's begin with fileutil. I wrote fileutil to give me more information about my files areas without having to look at the files areas manually. The options available are c, n, m, u and d. They are:

Option	Description
c	create files areas. For **INITIAL SETUP ONLY**. This option will scan the directories that are supposed to have files in them, and automatically add those files to the files area automatically (with blank descriptions). This requires a files area name (just to make sure you want to do this)

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

n List the new files in all files sections to a file. Requires a filename to save the list to.

m List the most popular files on the BBS. Requires a filename to save the list to.

u Update all files areas for new files and delete those files with zero length in their descriptions.

d Delete files that exist in the headers but do not exist in the files directory. You want to do this occasionally to clean out failed uploads and files you may have deleted. Note: You want to clean up the uploads area prior to executing this command. If you do not put the files found in the upload area in the files directory, the description for the file will be deleted.

Here is a sample output from a typical 'new files listing:

```
New files in the last 7 days.  (05/25/94)
Name           Section                    Date   by    Downloads
cyclone1.cpt   Macintosh Games          05/20/94 sruby     0
cyclone2.cpt   Macintosh Games          05/20/94 sruby     0
modvoicer1.1.c Macintosh Sounds         05/22/94 sruby     0
mono2stereo.cp Macintosh Sounds         05/22/94 sruby     0
2WayTalker2.1. Macintosh Sounds         05/22/94 sruby     0
99bottlesofbee Macintosh Sounds         05/22/94 sruby     0
note           Macintosh Utilities      05/20/94 sruby     0
eyeballs.sea   Macintosh Utilities      05/20/94 sruby     0
numbercrunch.s Macintosh Utilities      05/20/94 sruby     0
todo3.03.sit   Macintosh Utilities      05/20/94 sruby     0
windows3.0.cpt Macintosh Utilities      05/20/94 sruby     0
poor-mans-newt Macintosh Utilities      05/20/94 sruby     0
chemcalc.sit   Macintosh Utilities      05/20/94 sruby     0
clik'x.sit     Macintosh Utilities      05/20/94 sruby     0
periodictable. Macintosh Utilities      05/20/94 sruby     0
intrslip.sea   Macintosh Utilities      05/20/94 sysop     0
```

The 'most popular' listing is similar.

Typically, you'll run 'new' files, 'most popular' files and 'update' files options nightly via cron. Here is the crontab I use on the BBS to generate the reports automatically to system text files:

```
# bbs crontab -- automatic bbs maintenance
#
# update file sections for new files
0 3 * * *       setenv BBSDIR /bbs; /bbs/bin/fileutil -u
#
# update new files listing
1 3 * * *       setenv BBSDIR /bbs; /bbs/bin/fileutil -n
/bbs/text/system2.msg
#
# update most popular files listing
2 3 * * *       setenv BBSDIR /bbs; /bbs/bin/fileutil -m
/bbs/text/system3.msg
```

Note how the BBSDIR environment variable is set prior to executing the command. Mail is sent nightly to 'bbs' for the first command, and the output for the next two commands go to system2.msg and system3.msg so that users see the new files and most popular files upon logon.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

The next utility is a bit easier to digest. The errlogd program is the BBS error logger. You want to start this when the BBS is running. It generates no output; it collects error and status messages from BBS processes and places them in /bbs/admin/bbserr.

To start the program, use the following command line (as the BBS Administrator):

/bbs/bin/errlogd &

The above starts the error logger and pushes it into the background. See the below section for a more succinct description of the errlogd utility.

The last program is the most fun, and, the most underdeveloped. The 'monitor' program was a program that I wrote so that I could watch what users are doing on the BBS. This program contacts the BBS, and tells it to send all output to it as well as to the user. This is very handy, as you can see *EXACTLY* what the user is doing at any time. I use it to see where people are spending their time, to look for potential errors (and confusion) on the part of the user because of something that is unclear in the BBS (like a menu they can't figure out) and generally to see what people are doing on my BBS.

To run the monitor, simply type /bbs/bin/monitor. At that point, the monitor will wait for someone to log onto the BBS. Once they've logged on, the BBS is contacted to say "Hey, I'm here. Talk to me too!". At that point, you'll see what the user is seeing (with exceptions; when a user is looking at a text file via 'more', you'll see "displaying text file XXXXXXXX").

There is one problem with the monitor program. If you have multiple users using the BBS at the same time, the monitor won't work. Since the monitor is still in the development phase, it will talk to one BBS process only. If multiple BBS processes talk to the monitor, I don't know what will happen, but I doubt it will be good.

In the future, I am going to be adding to the monitor program. I want to turn it into the SysOp's BBS Console. This will allow the SysOp to monitor *any* user of the BBS, change a user's access level on the fly or even log out a user that is irritating the SysOp. (and many more items that I don't want to elaborate on at this point).

Watching for Problems

The rocat BBS system logs all messages and errors, if at all possible. However, to do this you must turn on the error logging daemon. This is done by logging in as the BBS Administrator, and typing the following command:

```
/bbs/bin/errlogd &
```

The above command will start the error logging daemon and push it into the background. The daemon will run in the background, waiting for messages from BBS programs about errors or status messages.

The file that is used for error logging is /bbs/admin/bbserr. You'll want to check this file FREQUENTLY in the first months of running the BBS. This file will tell you if you've got problems with menus, files areas, etc. It will also tell you who's logged in, and what they did (within reason). A sample output would be:

```
05/20/94 10:58:30: (S8) Logon for Alpha Tester
05/20/94 10:58:54: (S8) Got signal 1
05/20/94 10:58:55: (S8) (hangup) Logoff for Alpha Tester
05/20/94 12:54:19: (S8) Logon for Bill Clinton
05/20/94 12:55:03: (S8) Bad line A in files area macapps
05/20/94 12:55:03: (S8) Filename was bbedit
05/20/94 12:56:51: (S8) Logoff for Bill Clinton
05/20/94 21:47:20: (S9) Logon for Alpha Tester
05/20/94 21:47:45: (S9) Got signal 1
05/20/94 21:47:45: (S9) (hangup) Logoff for Alpha Tester
05/20/94 21:48:51: (S9) Logon for Alpha Tester
05/20/94 21:49:06: (S9) Got signal 1
05/20/94 21:49:06: (S9) (hangup) Logoff for Alpha Tester
```

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

The above list shows what goes on in the BBS. In the first few lines, you see Alpha Tester log onto the BBS on tty line ttyS8. (it was really me doing some testing). The BBS got signal 1 (a hangup, caused by turning off the modem), which caused the BBS to log the user off. Note that the BBS will always update it's information, if possible, upon logout of a user. (e.g. if the user decides to be tricky and download a bunch of files, and drop carrier in hopes of getting his time over again, it won't work)

Note that a hangup is not a graceful way to exit the BBS. You really don't want people doing this on a regular basis. It breeds a sense of irresponsibility in users, and tends to irritate SysOps.

Troubleshooting

Potential Problems

Most potential problems can be attributed to permissions problems. This is why I have written a shell script that should be run every night to check permissions.

Additional Information

File Formats

All files found within rocat are completely editable text. This allows you (the SysOp) to go into the file and look for (and hopefully fix) problems. This BBS is not at the 1.0 level yet; I don't feel it's 100% bug free. You will have corruption happen eventually. Whether it is a bad disk block, a panic of the system or even just a user with a weird combination of letters, you will see something wrong eventually.

There are two files that haven't been described yet, the userlog and a files header (there are many files headers in a typical BBS installation). Let's start with the userlog.

For every user in the BBS, there is a userlog record. The record is comprised of 4 lines of data, space separated. A typical line follows:

```
[A shaw Greg Shaw shaw CO Westminster ]
[B vt100 769579635 1 0 0 blue ]
[C 0 0 0 1 vi 24 80]
[D 0 200 40 8 769579635 0 ]
```

In the above, you see my entry. As you can see, there is a lot of information in the entry, and it's not obvious about what a lot of those 0's and numbers mean.

Before I jump in, I've labeled each line uniquely with A through D. Also, every line is within square brackets. I didn't add this when I put it into this document. This format is required for the BBS. This allows the BBS to check for errors in the Userlog file. Every record should have an A through D record. If that format is broken, an error message will be sent to the error logger, and the BBS will exit (to make sure that damage isn't propagated).

The line formats are:

[A login_name first_name last_name alias state city]
[B terminal_type last_logon_time #_of_logons downloads uploads card_color]
[C #_of_private_messages #_of_public_messages credited_time has_color editor lines columns]
[D flags access_level timelimit timeused_last_call anniversary_date K_downloaded_last_call]

The fields are (in order of ocurrence):

Field	Description
(line A)	
login_name	The name used to logon to the system (from the passwd file)
first name	The user's first name
last name	The user's last name
alias	The user's alias (to be used for chatting)
state	The state the user calls from

city	The city the user calls from
(line B)	
terminal_type	The terminal type that the user's software supports
last_logon	Unix time format (seconds) for when the user last logged on
#_logons	The number of times the user has logged into the system
downloads	The number of downloads by the user
uploads	The number of uploads by the user
card_color	The access card color of the user
(line C)	
#_private	The number of private messages the user has entered (not used)
#_public	The number of public messages the user has entered (not used)
credited_time	The number of minutes of credited time the user has available
has_color	Does the user have color? (0 for no, 1 for yes)
editor	The name of the user's favorite text editor
lines	The number of lines on the user's display
columns	The number of columns on the user's display
(line D)	
flags	The user's access flags
access_level	The user's access level
timelimit	The number of minutes the user gets per logon
timeused_last_call	The number of minutes the user used last call
anniversary_date	The date of the first logon by the user
K_downloaded	The amount of 1024 byte blocks the user downloaded last call

Some notes:

- The city field can be multiple words
- The login name can be only two words. The BBS warns users not to enter more than two words, or enter an underscore between the last two words.
- timeused_last_call is cumulative over the time period defined by WAITTIME in the config file. If the user logs in for 5 minutes at the top of each hour for four hours, the timeused_last_call will be 20. This is to make sure that users don't get more time than they're supposed to get.
- The credited time field is also cumulative over the WAITTIME period. After 12 hours, that time is discarded.
- Access level, card color and flags are independent of one another. (see above sections for description)

The other file format that hasn't been described is the files header file. A typical line from that file would be:

```
[A shaw 1 pb10.zoo]
[B A GIF/JPEG viewer for VESA systems. ]
[C DOS, VESA compatible driver ]
[D This is a GIF/JPEG viewer for DOS.  It uses VESA compatible modes, so if
you]
[E have a VESA compatible driver, get this program. ]
[F  ]
```

The files header information is comprised of 6 lines, of the same format as the Userlog information (square brackets and A-F labels). The contents of a record:

[A uploader number_of_downloads filename]
[B short description]
[C hardware/software requirements]
[D line 1 of long description]
[E line 2 of long description]
[F line 3 of long description]

Most of the above is self explanatory. A few notes:

- The filename must not have spaces in it at this point. I will be fixing this in the next release. I will be adding quotes around the filename.
- The short description should be less than 40 characters (so that it will fit on the screen)
- All fields from B onward are optional. Nothing will be printed if the field is empty.

How to connect to FIDONet

At this time, I do not have FIDONet installed on my BBS. I hope to do this in the near future. If you wish to do this, please look at the IFMail package available on the Internet.

UUCP, Mail and News

UUCP is the primary transfer medium for many e-mail and news sites. It is my primary medium. Please read the Linux HOWTOs and the Linux FAQs for instructions for installation and use of UUCP/News/Mail. These are really complex subjects.

Other useful programs

Some of the useful programs I have on my BBS available to BBS users are:

Program	Description
online games	Online games are basically Unix style games run through the BBS interface. They work remarkably well.
additional editors	The more editors on your system, the better chance a user has to edit a message with something that he (or she) has used previously.
sendmail	I've got sendmail setup on my system. I trust it more than smail, but that's personal preference. You really **need** a working mail system for a BBS.
cnews	The c-news package for handling the news system. Set it up and forget it, basically. It runs itself.
elm	An easy to use mail program with great help (and **excellent** features)

Of course, there are tons of other programs available for LInux (Unix). Add them as you wish; many of them are indespensible for this reason or that reason. (again, this is a personal opinion)

To Do

There are a number of things that I plan to do to continue developing the BBS interface. When I initially designed the BBS, I wanted to have a BBS with at least the capabilities of current BBS systems (within reason; I don't consider a full graphical user interface reasonable at this point). The two items that I wanted to have in my BBS that I didn't already have (or I simply wanted) were color/ANSI/curses and the multi-user chatting module.

ANSI/color/curses support is a major item in my mind. Color adds clarity to the BBS screens, and if curses is supported (and the terminal package is relatively accurate) you can have very involved screens and input.

There are a number of multi-user chatting packages available at this time. The real feature that I wanted to have by bundling the chatting package with the BBS was the capability for users to send each other messages, and, to instantly know what sort of population currently resides in the individual chatting rooms. If I can successfully port a BBS package (such as IRC) to the BBS and get permission from the author to include it with the BBS, I shall certainly do so. I do not wish to re-invent the wheel if I can possibly help it.

More than two serial lines

This BBS was meant to run with any number of lines/users (provided your computer can handle the load). However, I wanted to make sure that you understand what is related to using more than one serial line.

Generally, running more than two serial lines requires the use of either a Unix-specific card or a 'smart' card. The Unix specific card is what I currently use in The Roman Catacombs. I like its capabilities, and especially the price. These cards run about $100.00 (US dollars).

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

However, these cards can influence the load on the system because of the number of interrupts generated during high-speed file transfers.

This is why I recommend a 'smart' card for anything more than 4 lines. The only downside is that there are no 'smart' products currently available for Linux. I hear that some are in development (the drivers are under development; the cards are freely available). Hopefully soon they'll be available.

I know I have glossed over the above two items, but I don't want to re-iterate information that is available in the Linux Frequently Asked Questions and the Linux HOWTOs. For more information, please see these documents.

Multiple Machines and Large Setups

Monitoring through IP sockets

Let me explain a little bit more about how the BBS reports errors and does it's 'monitor' connections.

Currently, the BBS uses IP sockets as it's primary Inter-Process Communications medium (IPC). This allows you, the SysOp, to monitor the system from another location on the network. This is done by the changing two lines in the options (bbsinfo) file:

Option	Description
LOGHOST	the hostname of the host that is running the errlog daemon
WATCHHOST	the hostname of the host that is running the monitor daemon

If you should have two machines, you can change the above options to point to your other machine, if necessary. I wrote it this way for two reasons:

1. In my opinion (IMO), any BBS machine that runs this program (and doesn't have tons of horsepower) will be a dedicated machine.

2. Should you wish to run multiple BBS machines, you'd want all error messages to go to a single file. I keep this in mind because I've seen quite a few BBSs blossom into HUGE network based systems. A BBS that focuses on single-machine usage limits the growth of the BBS.

Now I may be wrong about number one above, but I like to plan for future growth rather than limit myself to single machine designs.

A few notes about running multiple machines:

- When running with multiple machines, you want your most reliable system to run as the error logging system
- Should you run multiple BBS machines, running one pointing to itself and the other pointing to the first is the best option.
- If you have two linux boxes and want to run the BBS on one and the error logging processes on the other, please remember that no error logging will be done should you, for instance, reboot to DOS to play a quick game of DOOM. In this case, you'll want error logging to stay on the BBS machine.
- If it wasn't clear before, running the error logging daemon is optional, but generally useful. It isn't necessary to run the logger, but if it is not running, it won't be able to tell you should an error occur.

Terminal Servers

Another option that I didn't discuss was the use of terminal servers as a front end to the BBS. Terminal servers are dedicated computers that interface with modems and connect from the modem to the target computer upon connection. This allows the computer to off load communications interrupts from the machine. All traffic comes (and goes) across the network.

At this point, I'd consider my dream BBS to be the following:

A number of BBS machines (top of the line X86 machines) with high-bandwidth busses.

Network (ethernet, probably) to communicate between the boxes.

A couple of terminal servers to handle the communications chores.

A direct connect to the Internet through a dedicated 64Kbps line or an ISDN line.

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

TONS of disk space (of course)
TONS of phone lines

This may not be especially practical, but I do like to dream!

The Roman Catacombs Setup

In this section, I'd like to describe exactly what my BBS looks like. This is to allow you, the potential SysOp, to see what I've done with the BBS and give you some ideas on how you'd like your BBS to look.

Basically, I've created a BBS that runs in the directories described in the previous sections. However, I've done a few things that aren't obvious in the previous discussions. In particular, I've got a somewhat hierarchical menu system with respect to the Linux files areas.

I currently support direct access to the Linux areas of the BBS without authorization or pre-checking. This allows someone to download the Linux disks that they require without running into upload/download limits and time constraints (within reason; I don't want them to download the entire set at one time, it would take too much time) Basically, I've got a Linux user. I've deleted his password so that no password is required. He has his own 'card', which allows 2500K of download per session, and unlimited download privileges. The Linux user can do only one thing. That is, he can download Linux disks, and that is it. Here is the main menu:

```
0|||||||  Roman Catacombs
Entry Vault
0|||||||  ------------------------------------------------------------------
-----------
0|||||||
0|||||||                                      Exit
0|||||||
0|||||||                                         <G>
0|||||||                                 ----------------------------
0|||||||                                 H>elp
0|||||||                            <W>  P>rivate Mail                <E>
0|||||||             Art Studio --  R>egistration Instructions -- Computer
Files
0|||||||                                 U>tilities
0| 400 >||||                               S<Y>sop's Area
0| ||||                                    C>hat with SysOp
0| 51 < |||                                  L>inux Files Areas
0| ||||                                    ----------------------------
0|||||||                                         <S>
0|||||||
0|||||||                                    News and Games
0|||||||
0|||||||  ------------------------------------------------------------------
-----------
0|||||||  <-> Previous Menu     <F> Feedback to SysOp   <G> Quick Logout
1|g|||||
3|-|||||
8|c|||||
2|y|400|>|||sysop|
2|e||||computerfiles|
2|w||||art_studio|
2|s||||newsngames|
2|u||||utilities|
2|l|50|<|||linuxtop|
21|f|||||elm -s feedback shaw|
2|p||||mail|
10|r||||reg.instr|
2|h|||||helptop|
```

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

The Linux user's access level is set to be 50. If you'll look at the line containing 'linuxtop', you'll see that anybody with access below 51 (50 or below) will see the 'Linux Files Areas' menu option.

When the user selects the Linux Files Areas option, he is transferred to the following menu:

```
0||||||| Roman Catacombs
Linux Top
0||||||| ------------------------------------------------------------
------------
0|||||||
0|||||||                            ---------------------------
0|||||||                            1. Slackware Distribution
0|||||||                            2. Linux Source
0|||||||                            3. Linux Utilities (source)
0|||||||                            H. Help
0|||||||                            ---------------------------
0|||||||
0||||||| ------------------------------------------------------------
------------
0|||||||  <-> Previous Menu     <F> Feedback to SysOp   <G> Quick Logout
1|g|||||
3|-|||||
8|c|||||
2|1|||||linux/linuxslackware|
2|2|||||linux/linuxsource|
2|3|||||linux/linuxutilities|
21|f||||elm -s feedback shaw|
2|h|||||helptop|
```

The operation that isn't obvious is the menu transfers at the bottom of the menu. Note how there is a slash in the menu title. If the user selects '1', he will be transferred to the menu named (with full path) /bbs/menus/linux/linuxslackware. This sort of operation allows you to nest menu directories based on subject, type of operation, etc.

Updating from rocat 0.75

There have been a number of changes since 0.75. They are:

- Bug fixes (of course!)
- Support for CD-ROM filesystems
- A change in the configuration of the files header files to eliminate duplicated information.

See the CD-ROM section for how to setup a CD-ROM section.

The files section header used to contain both the size of the file and the date the file was uploaded. I did this so that I wouldn't have to look up the file on every access.

Well, it turned out that I started doing that anyway so that I could indicate whether the file was available. Combining the two operations resulted in the file information being read from the directory structure 'live'. Note that this could impact a large files section (~500 files or larger) on a slow drive (aka CD-ROM).

Due to the change in the files header, you *MUST* update your files section headers to the new format prior to installing the program. I've furnished a program, filesupdate, that will do this for you. Simply run the program for usage (and *SAVE* a copy of your old sections, just for security).

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved

Conclusion

I really must apologize for the brevity of this document. It's a rush job. In the future, I want to put the following things into this document:

- A walk through for those who have never run a BBS before.
- More pointers on how to setup the BBS for Linux
- More examples

Well, that's where it has to end. This BBS isn't finished, but it works REALLY well. I think the BBS has some of the most powerful capabilities I've ever seen. This, coupled with the native programs and services available on Linux (Unix) make the BBS a VERY powerful system.

To Contact Me

PLEASE don't expect me to walk you through your installation. I've tried to explain how to setup the BBS as completely as possible.

If you want support, I expect payment for services rendered. I will offer service contracts, should you wish it. (and want to pay for the BBS program)

I expect you, the SysOp, to ask the Internet (through news) about problems with configuration and/or bugs prior to contacting myself. If I'm not very wrong, there will be a LOT of SysOps running this software in a short amount of time. I am very very very busy, and expect to be so in the near future (aka the next 10 years). If a question is either obvious (RTFM) or not on the topic of the rocat BBS system, I will send it to /dev/null.

If you get the BBS up and running, please drop me a note. I like to hear about success stories. Also, if you have any (reasonable) ideas, please drop me a note.

I wrote this BBS to be a **free** software package. I don't expect payment. However, should you find it in your heart (and wallet) to send me $25.00 (or a case of good beer), send it to:

fmSoft, Inc.
% Gregory Shaw
8060 Lowell Blvd
Wesminster CO 80030

I may always be contacted at my BBS:

The Roman Catacombs
(303) 429-8914
(v32bis, 24hrs)

And, of course, Internet mail:

shaw@manwe.fmsoft.com

The rocat BBS System, copyright © 1994 by Gregory Shaw and fmSoft, Inc. All Rights Reserved