

UNIX

January 26, 2005

Sequence Analysis & Consulting Service

Seminar Series

Susan Johns

(johns@cgl.ucsf.edu)

To work along during this seminar session, you need to have both a browser window open and be logged into your socrates account.

Log into your socrates account responding with your account name and password.

The items to be covered in today's seminar deal with the general operations of your SACS account and learning enough UNIX to accomplish your desired sequencing tasks.

account basics:

I have an account on socrates.

Socrates is the machine that houses the SACS sequence analysis software. This is a multi-user machine. Each user on socrates is given a set area on the machine in which to work called an **account**.

I have a SACS account.

A SACS account is supposed to be used for sequence analysis tasks. However, there are a lot of labs that just use SACS as a source of email accounts.

Your group leader has signed up for a SACS group account. Since SACS is a recharge unit at UCSF, this means that your group gets monthly bill for its SACS computing activities. Normally, the charge is \$100 a month for an unlimited number of individual accounts. Each individual account has an initial quota of 10 Mbytes. When the combined total of all the disk storage being used in a group exceeds 10 Mbytes, an additional disk storage charge is incurred (\$.005/KB).

How do I manage my account?

Because socrates uses UNIX as its operating system, to effectively manage and carry out sequence analysis task in your SACS account, you need to learn something about UNIX.

UNIX is not as standard as one would think due to its mixed origins. Each computer company seems to have its own flavor of UNIX. Which means that functions with the same name may perform differently on different platforms.

Managing your account requires that you gain the following basic computing skills:

- create, edit and remove files
- create and remove sub-directories
- move around in your account
- send email to others

Account limitations

There are limitations on accounts to insure that large numbers of individuals can all be using the machine at the same time without impacting each other's work.

One of these limitations, called a **quota**, restricts the amount a disk space available for an individual account to use. This is to insure that no one user can take over the entire disk capacity of the machine and prevent others from working.

Another limitation is a default limit of **one hour of CPU time** for any one computing task.

Both of these limitations can be gotten around when a user really needs more disk space or is running a CPU intensive task.

BASIC UNIX

UNIX has a help facility known as **man** pages. This function only works, however, if you know the name of the command that you are trying to use. The information given is also somewhat dry and at times is less than informative to the novice user. For UNIX information organized in a more user friendly manner, check out the following web site.

<http://www.sacs.ucsf.edu/Documentation/UNIXHelp/>

Some basic commands to answer a few basic questions.

```
whoami lists the username
date gives the current time
pwd tells current location on the machine
```

To use your account, you need to know what is in it and why it is there. To find out what is in your account, use various variations on the **ls** command.

```
ls list names of files
ls -a list names of all files
ls -l list names of files with complete information
ls -la list names of all files with complete information
```

example output from the **ls-la** command

```
socr:<13> ls -la
total 121
drwxr-xr-x 3 gallo pga 8192 Jan 21 15:47 ./
drwxr-xr-x 234 root wheel 24576 Jan 18 16:17 ../
-rw-r--r-- 1 gallo pga 0 Apr 3 2002 .addressbook
-rw----- 1 gallo pga 2285 Apr 3 2002 .addressbook.lu
-rwxr-xr-x 1 gallo pga 158 Aug 3 1998 .cshrc*
-rwxr-xr-x 1 gallo pga 262 Dec 14 2000 .login*
-rwxr-xr-x 1 gallo pga 110 Aug 3 1998 .logout*
-rw-r--r-- 1 gallo pga 106 Sep 15 2003 .ncbirc
-rw----- 1 gallo pga 16643 Nov 10 09:01 .pine-debug1
-rw----- 1 gallo pga 12682 Sep 22 12:12 .pine-debug2
-rw----- 1 gallo pga 12881 Sep 22 12:11 .pine-debug3
-rw----- 1 gallo pga 13564 Sep 22 12:15 .pine-debug4
-rw----- 1 gallo pga 16661 Nov 10 09:01 .pinerc
drwx----- 2 gallo pga 8192 Nov 10 09:01 mail/
```

File names that start with a dot are control files for your account and are usually not modified by the user. They can be with care, but be aware that you could greatly impact your account's operation by doing so.

```
.cshrc c shell control file
.login login control file
.pine* pine mailer control files
.address* pine mailer address control files
```

In UNIX the asterisk symbol, *****, is used as a wild card. UNIX is also **case sensitive**.

How can I look at the contents of a file?

To look at the contents of a file, use one of the following commands.

```
cat best used with small files
more best used with larger files
less similar to more, allows backward movement
```

cat can also be used to append files together to make longer files.

```
cat name1 name2 name3 > name4
```

In UNIX the greater than symbol, **>**, is used to

denote standard output, while the lesser than symbol, **<**, denotes standard input. The classic means of running a program in UNIX in given below using these two symbols.

```
program_name < input_file_name > output_file_name
```

To work effectively in an account usually requires the organization of the data files in a meaningful manner. One way to do this is through the use of sub-directories. Each sub-directory could contain the work being done on a specific project with additional sub-directories under that one to contain the various parts of the project.

```
                                -- blossom62
                                |
                                -- evolutionary ----- struct
                                |
cyto_c-project ----- heme-motifs
                                |
                                -- xray
```

sub-directory commands:

```
mkdir name creates a sub-directory
cd name moves you into a sub-directory
rm -r name removes a sub-directory and all its contents
```

Another means of organization is to use consistent naming conventions. Organize your sequences by data type and have each type end with the same extension. Name the output files from a given program with an identifying extension.

How can I work with files?

To change a file, use the following commands.

```
pico filename edit file content with the pico editor
mv filename1 filename2 change a file's name
cp filename1 filename2 copy file contents to another file
rm filename remove a file
```

Editing is an effective means of changing the contents of a file. The pico editor is a simple editor that is part of the pine mailer. The **vi** editor comes with all UNIX systems. It is very powerful, but has a huge learning curve.

Editing is a necessary skill to have when using a computer regardless of the software you use. By being able to create text files, you can more effectively control computing activities, use a greater variety of software, modify input and

output files, and communicate with others. Various editors are available on socrates, pick one and learn how to use it.

PICO is Pine's message composition editor. PICO is a very simple and easy-to-use text editor offering paragraph justification, cut/paste, and a spell checker.

Your SACS account also provides you with an email address if you care to use your account in that manner. One of the mailer programs on socrates is **pine**.

How can I search for a specific term?

The **grep** function allows file(s) to be searched for a desired term. This is a handy means of checking the contents of files to locate one of interest when you have forgotten the file's name.

```
grep seq .*
```

example output:

```
socr:<xx> grep seq .*
.cshrc:# This is the minimal standard .cshrc file for sequence
.cshrc:# The next line is necessary for sequence software
.cshrc:source /usr/local/lib/seq/cshrc
.login:# This is the minimal standard .login file for sequence softwa
.login:# The next line is necessary for the sequence analysis softwar
.login:source /usr/local/lib/seq/login
.login:setenv STADENROOT /home/socr/c/seqsoft/staden/staden_alpha
.logout:# This is the minimal standard .logout file for sequence ana
.logout:source /usr/local/lib/seq/logout
.ncbirc:BLASTDB=/mol/seq/blast/db
.pine-debug1: no-save-aggregates-copy-sequence
.pine-debug1: no-save-aggregates-copy-sequence
.pine-debug1: no-save-aggregates-copy-sequence
.pine-debug2: no-save-aggregates-copy-sequence
.pine-debug2: no-save-aggregates-copy-sequence
.pine-debug3: no-save-aggregates-copy-sequence
.pine-debug4: no-save-aggregates-copy-sequence
```

How can I change my password?

To change your password, use the following screen trace as your guide.

```
socr:<xx> passwd
```

You are registered with the following security mechanisms

- 1 KRB5
- 2 OSFC2

```
[Default selection: 2]
```

```
Select ONE item by number: 2<rtn>
```

```
You have selected:
OSFC2
```

```
Old password: (enter your old password)
Last successful password change for gallo: UNKNOWN
Last unsuccessful password change for gallo: NEVER
```

```
Do you want (choose one option only):
```

- 1 Pronounceable passwords generated for you
- 2 A string of characters generated for you
- 3 A string of letters generated for you
- 4 To pick your password

```
Select ONE item by number: (enter 4)
```

```
You have selected:
To pick your password
```

```
New password: (enter a new password)
Re-enter new password: (re-enter that new password)
```

Your changed password is then recorded throughout socrates.

What computing resources can my account use?

Since your account is a SACS account, it can make use all the software resources that SACS has available for its users. There are many different assignments necessary in order for all these programs to run. Therefore, your account has been automatically setup to have all these assignments made for you each time you log in.

To get an idea of the type of foundation that your account uses, enter the following command.

```
env
```

The **env** command displays the current environment variables associated with your account.

SACS software is spread throughout the socrates. Your account's path variable has been adjusted so that it can find and use all of the SACS tools.

The command **setenv** can be used to set any of the variables given in the env list. The following command would set the PRINTER variable in your account to cys, if entered at the machine prompt or placed in your .login file.

```
setenv PRINTER cys
```

ADVANCED UNIX

How can I see what's happening on socrates?

Socrates is a multi-user platform. Many people are using it at the same time, each wanting a part of the machine to do their individual computing tasks. Here are some commands to see who else is on the machine and what they are doing.

spying commands:

`who` lists which users are on, what terminal connection is being used and what time they logged in

`w` lists which users are on, what terminal connection is being used, what machine they are using, when they logged in, how long the connection has been idle and what they are doing

`top` lists the top CPU users on the machine

How do I check to see just how many things I have running on socrates?

This is done with the `ps` command. It allows you to check to see if you have hung processes or other computing tasks that you may want to stop.

`ps -u username`

example output:

```
socr:<xx> ps -u gallo<rtm>
  PID TTY          S       TIME CMD
 1681682 pts/32    S           0:00.15 -tcsh (tcsh)
```

If it is necessary to stop a process, use the process id (in the PID column) found with the previous command and the `kill` command to do it.

`kill -9 pid`

Can I string commands together?

Commands can be strung together by means of the pipe function denoted by the `|` symbol. The results of the first function are passed on to another for additional processing. In the example below `cat` is used to look at the contents of a file which is then handed over to the `wc` command with the `-l` switch to get the number of lines contained in the original file.

```
socr:<xx> cat .addressbook.lu | wc -l
      205
```

What else can I do with my account?

There are times when a user might have an account on another machine or need to move data from one place to another.

`telnet spin` start a telnet session on spin

`telnet ribozyme.vadms.wsu.edu` start a telnet session on another machine giving the complete IP address

`ftp ncbi.nlm.nih.gov`

start a ftp session at NCBI

Most ftp servers with information available to public have an anonymous ftp site. This means that access is gained through the use of the term **anonymous** as the account name and by giving your email address as your password. Most of these sites use the UNIX operating system and will allow you to use the `cd` command to move around once there.

You can use the `talk` utility to have an interactive conversation with another user currently on the machine.

`talk username`

Can I use x-term software on socrates?

Yes, if you have x emulation software already running on your local machine when you connect to socrates. You will need to have your machine set up to allow other computers to send x displays to your local machine. Depending on your machine's software, you may need to make your connection to socrates from that xterm window. You may also need to set the `DISPLAY` variable to have graphics displayed on the machine you are currently using. First, determine the name of the machine that you are using. Then set the `DISPLAY` variable.

`w | grep username`

`setenv DISPLAY name.domain.ucsf.edu:0`

Start up the desired x-term application on socrates and it should work.

When you terminate your current socrates session, be sure to close the xterm window. This is done by clicking in the window to activate it, and then entering `exit` at the prompt.

UNIX PRO

As a user starts to do more complex tasks, the ability to run tasks in the background or at particular time becomes important.

There are times when the desired task is the search of a database, a series of databases, or just a task that will take a long time. When

doing such a job, it is best to have it run in the background.

Normally such a process would be set up to use the batch queue, however, this doesn't work on socrates. Therefore, a user has really only one option. That is to create an input file that contains all the responses to the prompts that the program expects and use this as the input file to the desired process.

The best way to figure out what to put in such an input file is to run through the program a number of times with dummy data to determine what all the necessary responses are.

After the input file works successfully on a small data set, it is modified to do the desired task. Since a long time is required to complete the job, the user should enter **unlimit** at the prompt to insure that they have all the time they need for the job to run. After the environment to run the task is established, then enter a command similar to that given below to start the job.

```
program_name < input_file > output_file &
```

At this point, the user can log out or move on other tasks while the job runs along in the background. **top** can be used to check on the progress of the task if it is going to take days to complete.

UNIX offers the user the ability to run jobs at scheduled times through the use of the **crontab** utility. The type of tasks to be run in this manner is up to the user and can be as simple complex as needed. Consult with the man pages on crontab for the format required and the means of establishing such process. Once established, a crontab process will keep on running until the user stops it.

Where are things kept on socrates that I might use?

Socrates is organized with different areas for different functions. All the SACS accounts are in one or the other of these two locations.

```
/home/socr/c/users  
/home/socr/c/users2
```

The SACS secure web pages outputs results to the following locations depending which program was run.

```
/usr/tmp/webgcg  
/usr/tmp/phylicp  
/usr/tmp/autoblast
```

If you have a SACS supported web page, it is located in the following location.

```
/usr/local/html/sacs/home/username
```

Temporary scratch space is located in **/usr/tmp**. There is no limit on the size of the data to be kept here, only the length of time it is allowed hang around (two weeks). Touch can be used to modify time stamps and keep the data around.

If a user needs to have a large amount of data on-line for a long period of time and doesn't want to get billed for the disk storage, a user database area can be established in which to store their information.

Many novice users consider PERL to be part of the UNIX operating system. **It is not**. It is a stand alone scripting language that can be used on a number of platforms, including those using UNIX as their operating system. Becoming proficient in PERL requires that a user makes the effort to learn this scripting language and applies that knowledge to the solving of desired computing task. Running PERL on a UNIX platform will cause the code written to solve a given task to be written just a little differently than that used to do the same task on a nonUNIX platform.

UNIX is a complex and powerful operating system. Most of the time, a normal user doesn't need to know very much about its operation in order to work successfully on a platform that uses it. It is the job of the computer's system administrator to keep the computer and its operating system running smoothly.

If you want to know more about UNIX, there are numerous books available on the subject. They are written for various levels of UNIX experience.

URLs used in this seminar:

SACS UNIX Help for Users:

<http://www.sacs.ucsf.edu/Documentation/UNIXHelp/>

Pico help:

http://www.sacs.ucsf.edu/Training/introgcg/pico_help.html

Pico instructions:

http://www.sacs.ucsf.edu/Training/introgcg/pico_instructions.html

Pine instructions:

http://www.sacs.ucsf.edu/Training/introgcg/pine_instructions.html

Pine home page:

<http://www.washington.edu/pine>

