



UNIX PROGRAMMING





Logging In and Out

- **Login:** All users must provide a user name and a password to log into the system. Below is a dialogue:

```
login: stud01
```

```
stud01's Password: (password does not echo)
```

```
$
```

- In this case, the login has been demonstrated with my user name of stud01. The dollar sign prompt represents the shell (default of ksh in AIX).
- **Logout:** The logout command can be invoked with any of the following commands:

```
$ <ctrl-d> (press ctrl key and letter d together)
```

```
$ exit
```

```
$ logout
```

```
login:
```

- The logout commands will remove the user from their current shell. If the current shell is the only session, the result will be a new login prompt as above.





Invoking the Shell and the passwd command

- Type the name of the shell to invoke a new shell. The C shell appears as a % at the login prompt by default. Below is a dialogue:

```
$ ksh
```

```
$ csh
```

```
% <ctrl-d>
```

```
$ exit
```

```
$ logout
```

```
login:
```

- To create or change the password for the account, use the passwd command. Below is an example:

```
$ passwd
```

```
Changing password for ds59478
```

```
stud01's Old password: (no echo to terminal)
```

```
stud01's New password: (no echo to terminal)
```

```
Enter the new password again: (no echo to terminal)
```

```
$
```





Who Else is on the System?

- With the **who** command, you can see who is on the system and some information about each user:

```
$ who
stud01      pts/1          Dec 26 14:39
stud11      pts/9          Nov 17 10:27
$
```

- The output of the **who** command is the username, terminal information, their login date/time and the network address of the login point.
- The pipe operator (**c1 | c2**) takes the output of one command (**c1**) and uses it as the input of another command (**c2**). The command **wc** counts the number of lines, words and bytes in from the input. The **-l** option on **wc** gives just the number of lines. Therefore, we can combine the **who** and **wc** (with **-l** option) programs with a pipe to get the number of users on the system:

```
$ who | wc -l
    2
$
```

- We will cover the pipe as well as other operators in more detail in later sections.





Date and Time Functions

- With the **date** command, the shell returns the current date and time as below:

```
$ date
Sun Dec 26 15:05:40 EST 2003
$
```

- Note that there is a **time** command, but it performs a totally different function. The time command returns the amount of CPU time that it takes to execute a command. As you can see below, the CPU time to execute the command is negligible:

```
$ time

real    0m0.00s
user    0m0.00s
sys     0m0.00s
$
```





Month and Year Functions

- With the **cal** command, the shell returns the specified month or year:

```
$ cal 1 2000
      January 2000
Sun  Mon  Tue  Wed  Thu  Fri  Sat
    1
  2   3   4   5   6   7   8
  9  10  11  12  13  14  15
16  17  18  19  20  21  22
23  24  25  26  27  28  29
30  31
$
```

- To get the output for an entire year, leave out the month parameter:

```
$ cal 2000
<Try it to get the output>
```

- **Note:** Try the cal command for September 1752.
- Can you explain the output?





Other Simple UNIX Commands

- With the **clear** command, the user clears the screen:

```
$ clear
```

- The **echo** command writes to the screen:

```
$ echo Out to Lunch
Out to Lunch
$
```

- The **banner** command writes to the screen in large letters:

```
$ banner Out to Lunch
```

```
#####
# # # # #####
# # # # #
# # # # #
# # # # #
# # # # #
##### ### #
```

```
##### ###
# # # #
# # # #
# # # #
# # # #
# # # #
# # # #
```

```
#
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
#####
##### ### # # ### # #
```

```
$
```





Editing Ordinary Files – The vi editor

- Although there are a variety of editors available on UNIX systems, it is important to learn vi for the following reasons:
 - It is a standard editor available on all UNIX systems
 - It is the default editor for many programs
 - It is the only editor available in some situations
- The vi editor is an extremely robust program that is rich in features, options and customizability. In this course, we will cover only the most basic features of vi to build a working knowledge from which we can edit files.
- The vi editor works in two modes: command and edit. The command mode is the default when vi is initiated as follows:

```
$ vi file1
```
- Once vi is initiated, the following keystrokes will transfer to edit mode:
 - i – inserts text at the cursor
 - I – inserts text at the start of the line
 - a – appends text after the cursor
 - A – appends text after the end of the line





Command Mode – The vi editor

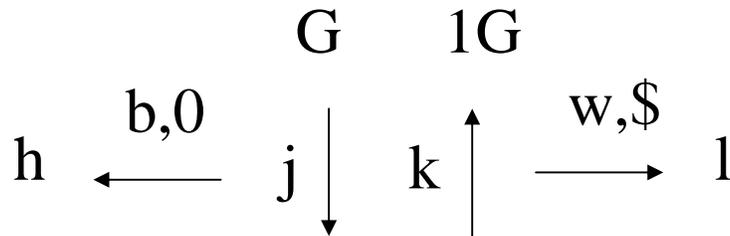
- To transfer from editing mode to the command mode, use the escape (<esc>) key. Once in command mode, keys control navigation as follows:
 - h,b,0 ← left, beginning of word, beginning of line
 - j,G ↓ down, end of document
 - k,1G ↑ up, beginning of document
 - l,w,\$ → right, end of word, end of line
- Text editing can be done with the following commands:
 - r – replace letter with a new letter
 - cw – change word (replace word with new word)
 - dw – delete word
 - x – delete letter
 - . – repeat the last command (u will undo the last command)
- Commands for the file are preceded with a colon:
 - :q! – quit out of the file without saving
 - :x,:wq, ZZ – exit and write (or save) the file
- Searches for can be done with the /<text pattern> command:





Summary – The vi editor

- Directional keys



- Edit mode
 - **i** – inserts text at the cursor
 - **I** – inserts text at the start of the line
 - **a** – appends text after the cursor
 - **A** – appends text after the end of the line
- Command mode
 - **<esc>** key places you in command mode
 - **:q!** – exit without saving
 - **:x, :wq, ZZ** – exit and save
- To correct mistakes
 - **x** – delete letter
 - **dw** – delete word





Customizing the Environment – editing ~/.profile

- By using the vi editor to edit UNIX files, we can learn how to customize the user environment by editing up the ~/.profile file. Follow these instructions carefully:
 - Bring up the .profile file in vi:

```
$ vi ~/.profile
```
 - Browse the file using the arrow keys. Do not make any changes to the file at this time.
 - Press the <esc> key to initiate command mode and enter the command to exit without writing to the file:

```
:q!
```

