1. Filesystem Tidbits:

   `/` is considered the "root" of the file system

   `.` is the current directory

   `..` is the parent directory

   `/home` is where the user home directories are stored.

   `~` will expand to your home folder path
   Example:
   `~/Desktop` is the same as `/home/smarinel/Desktop`

2. `pwd`:
   Prints the present working directory.

3. `ls`:
   The `ls` command lists the contents of the specified directory, by default the present working directory. Here are some common flags.

   `-a` : shows all files, including hidden ones

   `-l` : list contents with details

   `-h` : used with `-l`, displays file sizes in human readable form (e.g. K, M)

   `-X` : sort listing by file extension

4. `mkdir`:
   Simple, makes a directory of the given name.

   `-p` : creates the parent directories if they do not exist.

5. `cd`:
   `cd` changes the current directory to the one specified, truly exciting!

   `cd -` : This will change to the previous directory, good for moving back and forth

6. `cp` and `mv`:
   Copy (`cp`) or move (`mv`) the specifed file to a new location. Note `mv` is also used to rename files.

   Example: `cp TuxRocks.c ~` [Copies `TuxRocks.c` to my home folder]

7. `rm`:
   `rm` deletes the specifed file. IT CANNOT BE UNDONE, so be careful with rm.

   `-r` will delete recursively, such as when you want to delete a directory

   `-f` forces the delete.

8. `cat` & `tac`:
   Write the contents of a file to the screen, `tac` just prints lines in reverse.

   Print several files by doing: `cat file1 file2`

9. `head` & `tail`:
   Display the first (`head`) or last (`tail`) lines of a file. Default line count is 10, but it can be changed with `-n`

   Example: `ps -aux | head -n 20` (See Pipes for what the `|` means)

   By using the `-f` flag, `tail` will continue to output the end of the file as it is created. Very useful for following debugging statements outputed to a file, or for following logfiles. Press `Ctrl-C` to end the output.

   Example: `tail -f output.txt`

10. `man [section #] <topic>`: All things good and mostly true can be found in man pages.

    Important Sections:

    1 - General User Commands

    2 - System Calls

    3 - Programming Functions

    See less for a summary of movement commands in man

11. redirection:
    `>` Redirects output to STDOUT (`printf` etc) to a specified file.
    `<` Redirects input from a file to STDIN.
    `>>` append instead of write to specifed file

```
Example:
superseth@Pool-of-Tears:~/480/BotEnv/Agents $ ls -l > list
superseth@Pool-of-Tears:~/480/BotEnv/Agents $ cat list
total 24
-rwxr-xr-x   1 superset  superset  2090 Oct 15 17:57 MyAgent.class
-rwxr-xr-x   1 superset  superset  2900 Oct 15 17:57 MyAgent.java
-rwxr-xr-x   1 superset  superset  1641 Oct  2 17:11 NodeInfo.java
-rw-r--r--   1 superset  superset     0 Oct 15 18:16 list

cat file1 file2 >> file1 to copy file2 into file1
```

12. `pipe`:
    Creates a connection between two processes, sending the output of one to the input of another

    Example: `ls -l | less` : will show all files in directory one page at a time.
    `who | sort` : to sort the list of users currently on the system.

13. `grep`:
    Grep is a very powerful search tool using regular expressions, but to keep things simple here are some basic uses.

```
superseth@pool-of-tears:~/CSC/HBun/svn/src $ grep printf -n  *.c
9374:   sprintf(extStr,"%s%s",platExtStr,glExtStr);
```

    Here `grep` looks for the string `printf` in all files ending in `.c` in the current directory. `-n` lists the line number for each hit.

    Grep is also useful for figuring out how far into a man page something is
    `man mplayer | grep seek -n`

    See the man page or the web for more info.

14. `less`:
    Prints text to the screen, but allows scrolling using an interface like `man`.

    Useful Keys:

    U = Up

    D = Down

    Space = PageDown

    / to search

    Q = Quit

15. `top`:
    Lists information about currently runing progams. Press `Q` to quit.

16. `gcc`:
    `gcc` is the GNU C compiler

    Example: `gcc -g -Wall --ansi --pedantic TuxRocks.c -o tux`

    `-g` compiles with debugging flags, for use in GDB

    `--pedantic` checks for strict conformity to ISO C Standards

    `--ansi` checks for conformity to ANSI C Standards

    `-o <File>` specifies the name of the executable to output

17. `gdb`:
    The GNU debugger, very useful for working with C code.

    Some Commands:

    `r` – runs the program

    `bt` - prints a backtrace

    `p <var name>` – prints the value of the given varible

    `disp <var name>` – same as p but displays given variable every time you stop

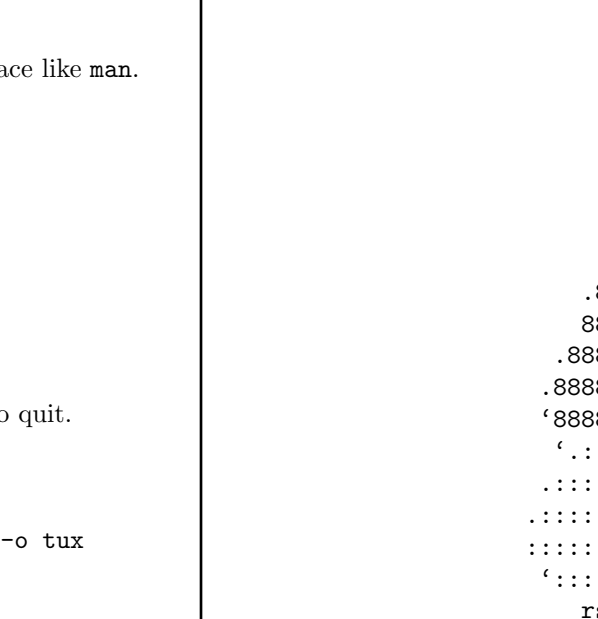
    `q` – quits

    `b <line number>` – sets a break point

18. `ssh userid@host`:
    Used to login to a remote machine.

    Related commands: `scp`, `sftp`

    There is a lot more, search online or see the links on our site.



`http://www.cplug.org`