Linux\_based\_commands

2. Problem Statement

Explain the below linux commands with an example. Share the screenshot of each

command description:

1. pwd - **s**hows the name of the current working directory

2. vi – start The default editor that comes with the UNIX operating system is called vi (visual editor). [Alternate editors for UNIX environments include pico and emacs, a product of GNU.]

3. touch - To simply create a blank file with touch command, use the syntax below.

4. mkdir -  make directory - command allows the user to make a new directory. Just like making a new directory within a PC or Mac desktop environment, the mkdir command makes new directories in a Linux environment. An example of the mkdir command

5. rm - remove - like the rmdir command is meant to remove files from your Linux OS. Whereas the rmdir command will remove directories and files held within, the rm command will delete created files. An example of the rm command:

6. ls  List files in the current working directory except those starting with . and only show the file name

7. echo - used in scripting language and batch files to display a line of text/string on standard output or a file.

8. cat - Sends file contents to standard output. This is a way to list the contents of short files to the screen. Here is an example - cat .bashrc (cat ~/.profile on mac).

9. who - Find who is on the system

https://www.cyberciti.biz/faq/unix-linux-who-command-examples-syntax-usage/

10.cd - Changes the current working directory . For e.g cd /home - this command will change your working directory to /home. The '/' indicates path relative to root.

11.date - is used to print out, or change the value of, the system's time and date information.

https://www.computerhope.com/unix/udate.htm

12.cal -  To display a calendar.

https://www.tutorialspoint.com/unix\_commands/cal.htm

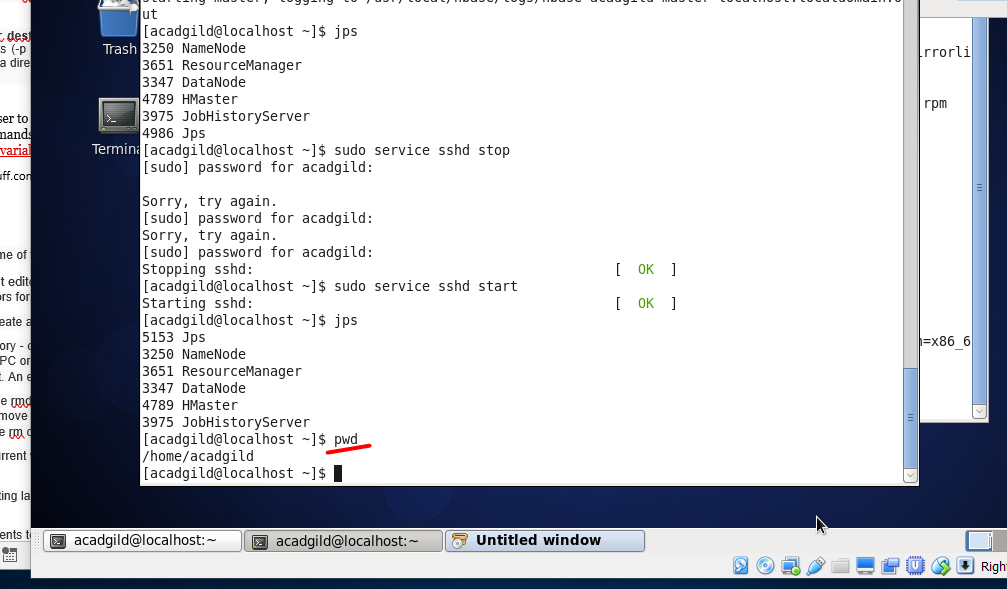
* 13.mv - **mv myfile yourfile :-**Move the file from "myfile" to "yourfile". This effectively changes the name of "myfile" to "yourfile".
* 14.cp - **cp myfile yourfile :-**This command copies the file "myfile" to the file "yourfile" in the current working directory. This command will create the file "yourfile" if it doesn't exist. It will normally overwrite it without warning if it exists. In order to get a prompt before overwritting, add a "-i" option like this to your command - **cp -i myfile yourfile.**Also, in order to copy a file from some location to your current directory you can do this - **cp -i /some/other/location .**
* **cp -dpr srcdir destdir :-**Copy all files from the directory "srcdir" to the directory "destdir" preserving links (-p option), file attributes (-p option), and copy recursively (-r option). With these options, a directory and all it contents can be copied to another directory.

15.which -  It allows user to pass several command names as arguments to get their paths in the system. “which” commands searches the path of executable in system paths set in $PATH [environment variable](http://www.thegeekstuff.com/2012/07/linux-export-command-examples/).

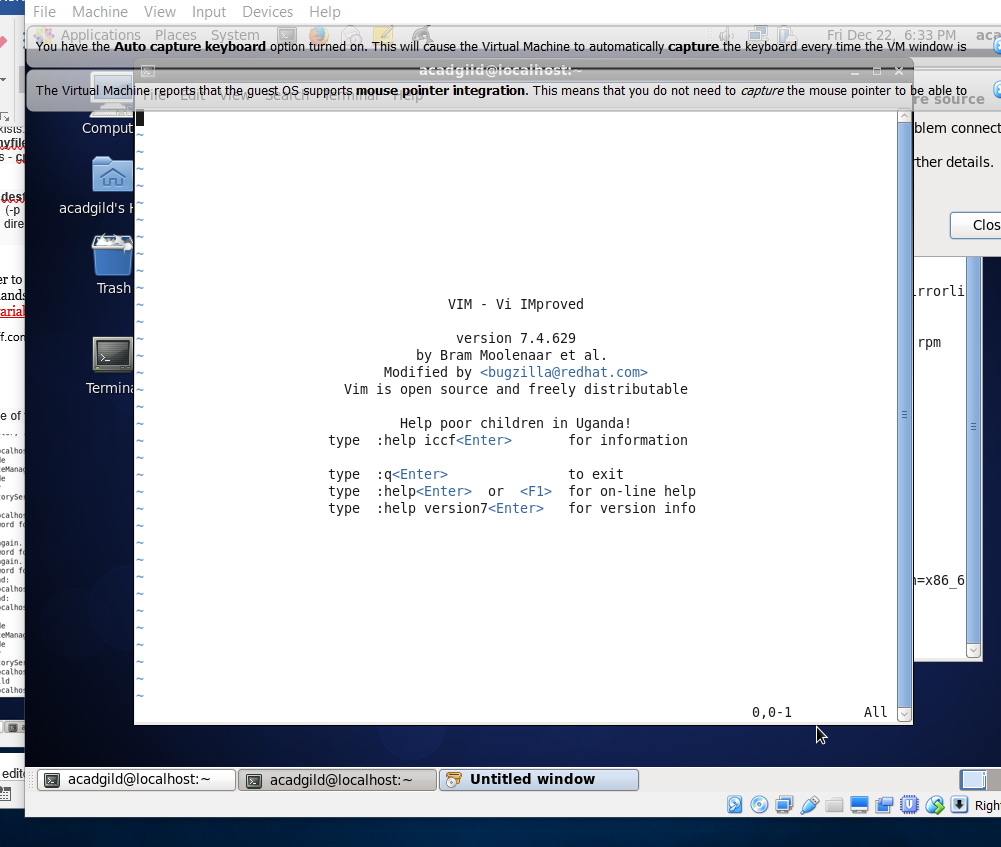
http://www.thegeekstuff.com/2013/04/linux-which-whatis-whereis/

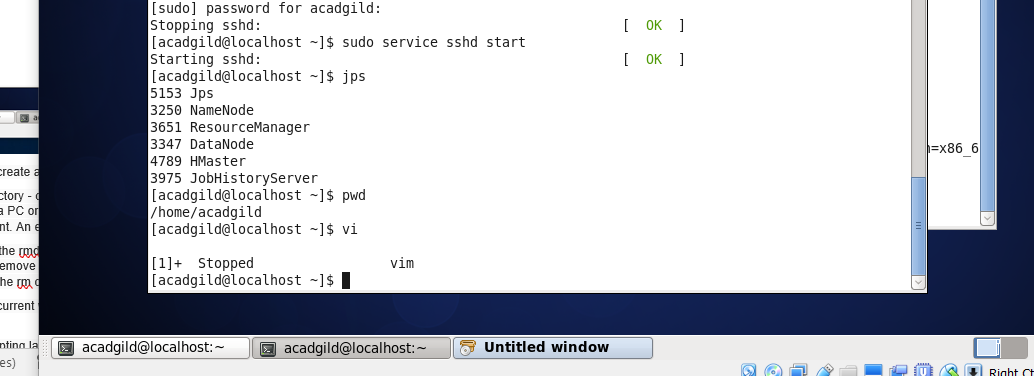
**3. Output**

1. pwd - **s**hows the name of the current working directory

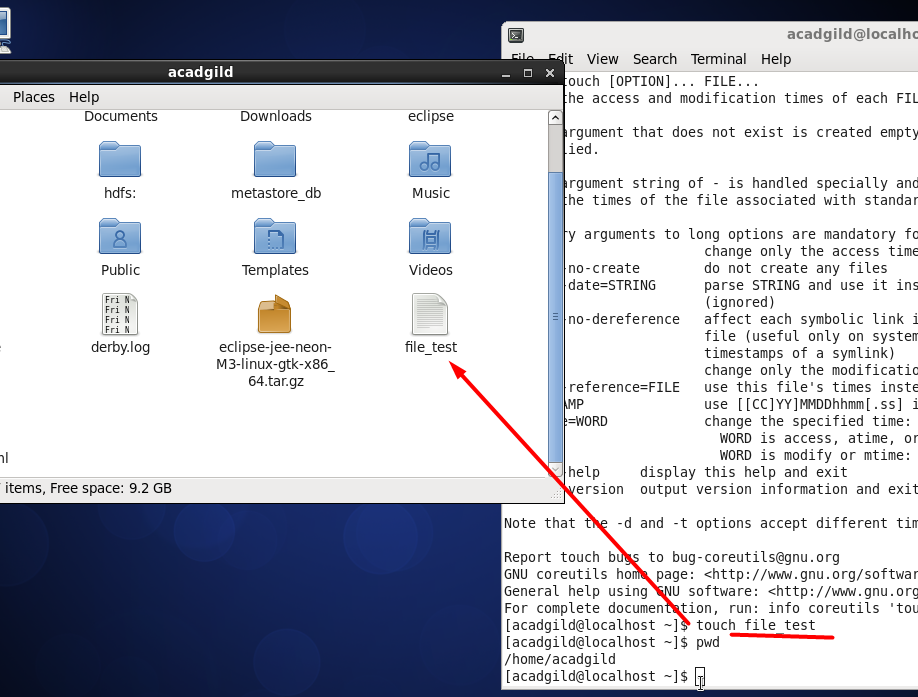


2. vi – start The default editor that comes with the UNIX operating system is called vi (visual editor). [Alternate editors for UNIX environments include pico and emacs, a product of GNU.]

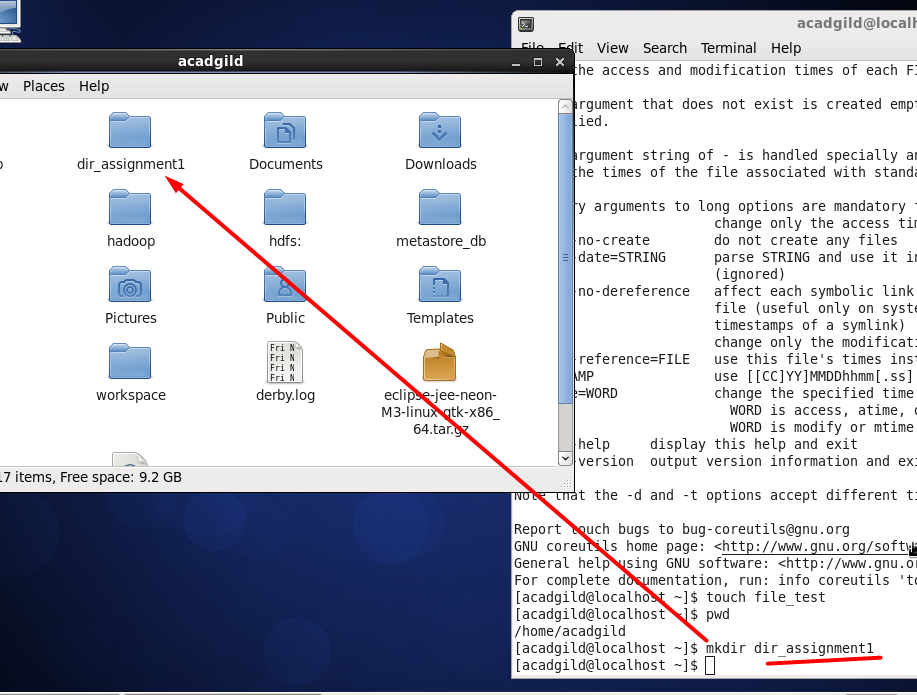




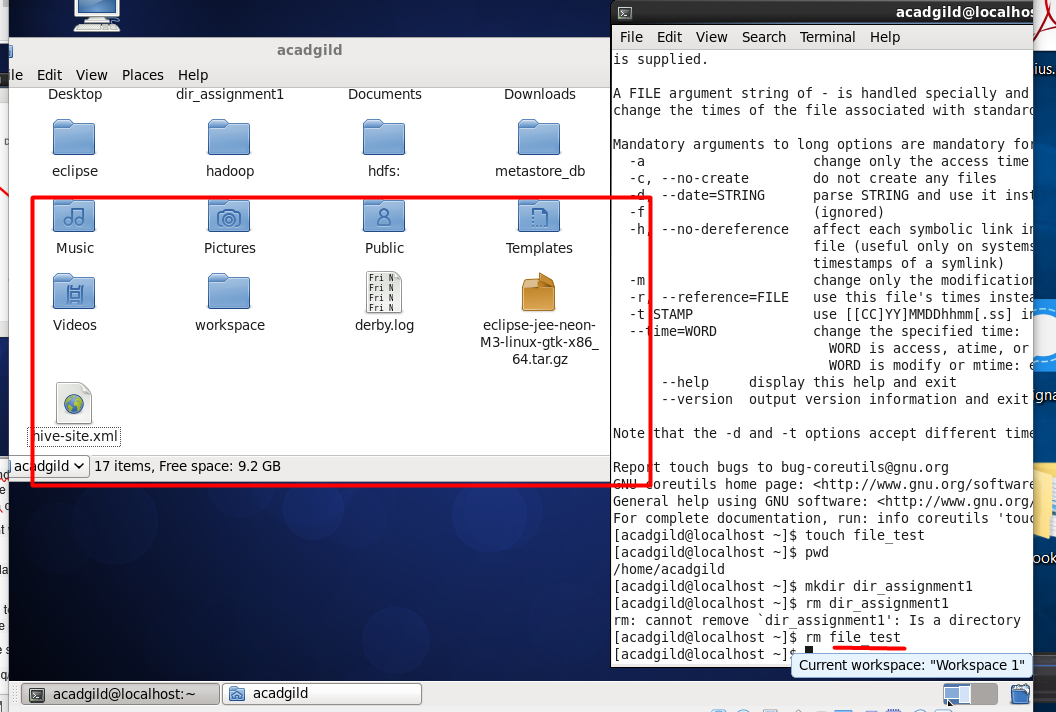
3. touch - To simply create a blank file with touch command, use the syntax below.



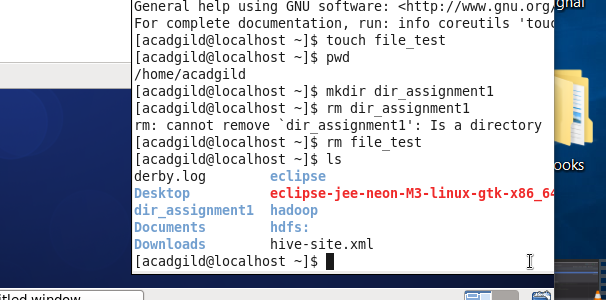
4. mkdir -  make directory - command allows the user to make a new directory. Just like making a new directory within a PC or Mac desktop environment, the mkdir command makes new directories in a Linux environment. An example of the mkdir command



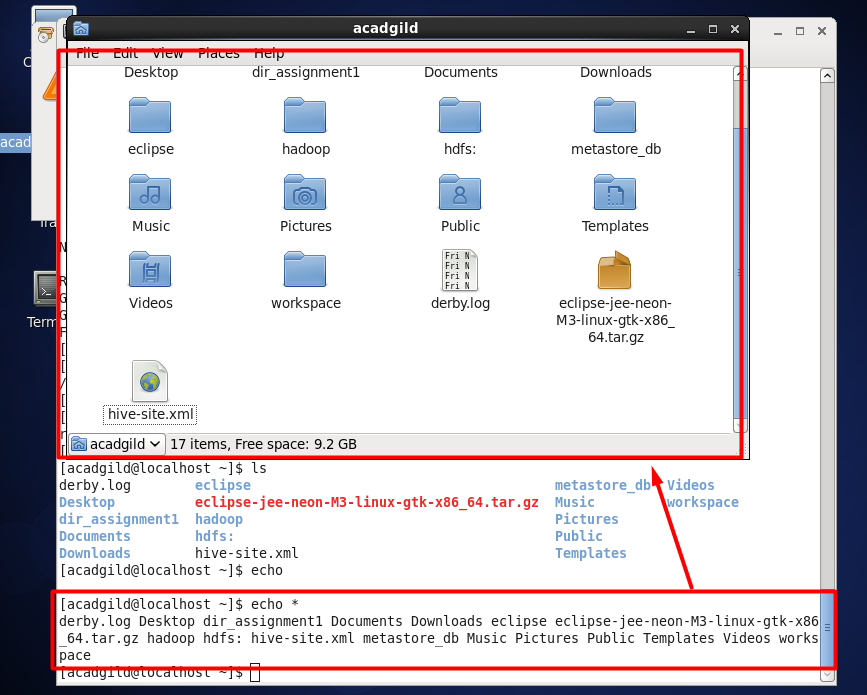
5. rm - remove - like the rmdir command is meant to remove files from your Linux OS. Whereas the rmdir command will remove directories and files held within, the rm command will delete created files. An example of the rm command:



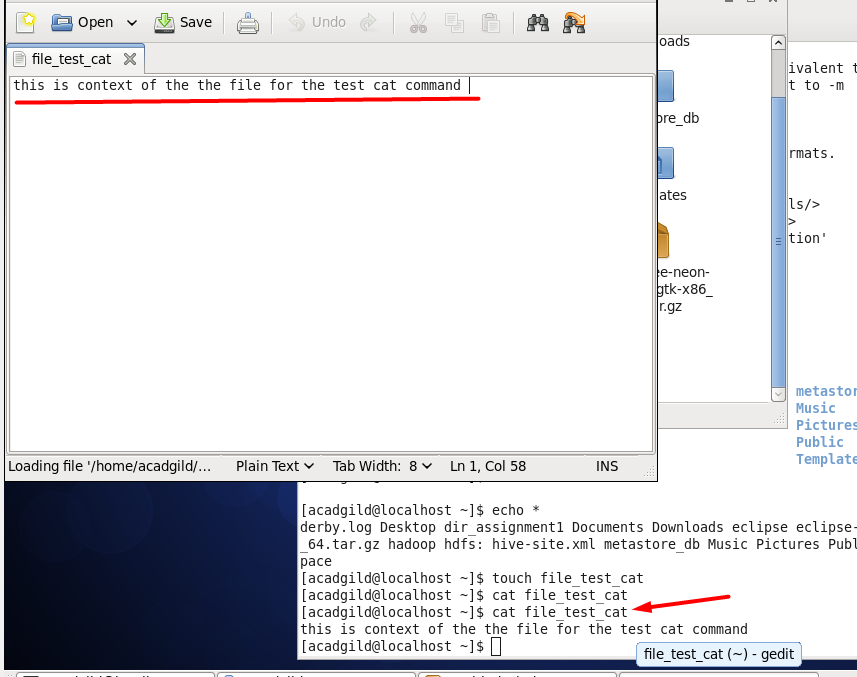
6. ls  List files in the current working directory except those starting with . and only show the file name



7. echo - used in scripting language and batch files to display a line of text/string on standard output or a file.

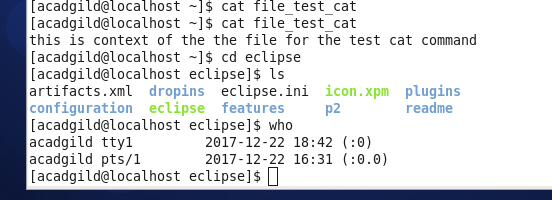


8. cat - Sends file contents to standard output. This is a way to list the contents of short files to the screen. Here is an example - cat .bashrc (cat ~/.profile on mac).

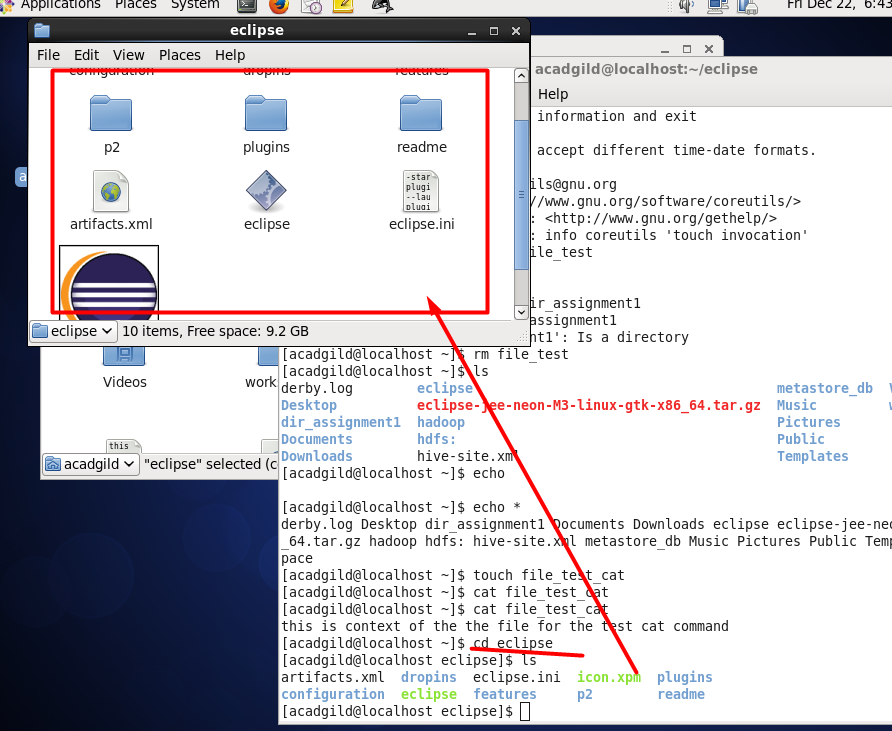


9. who - Find who is on the system

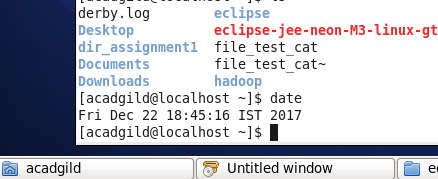
<https://www.cyberciti.biz/faq/unix-linux-who-command-examples-syntax-usage/>



10.cd - Changes the current working directory . For e.g cd /home - this command will change your working directory to /home. The '/' indicates path relative to root.



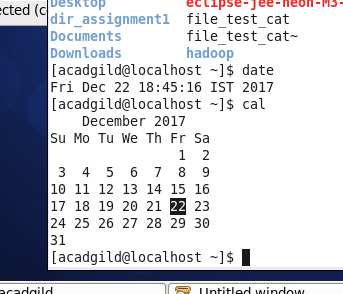
11.date - is used to print out, or change the value of, the system's time and date information.



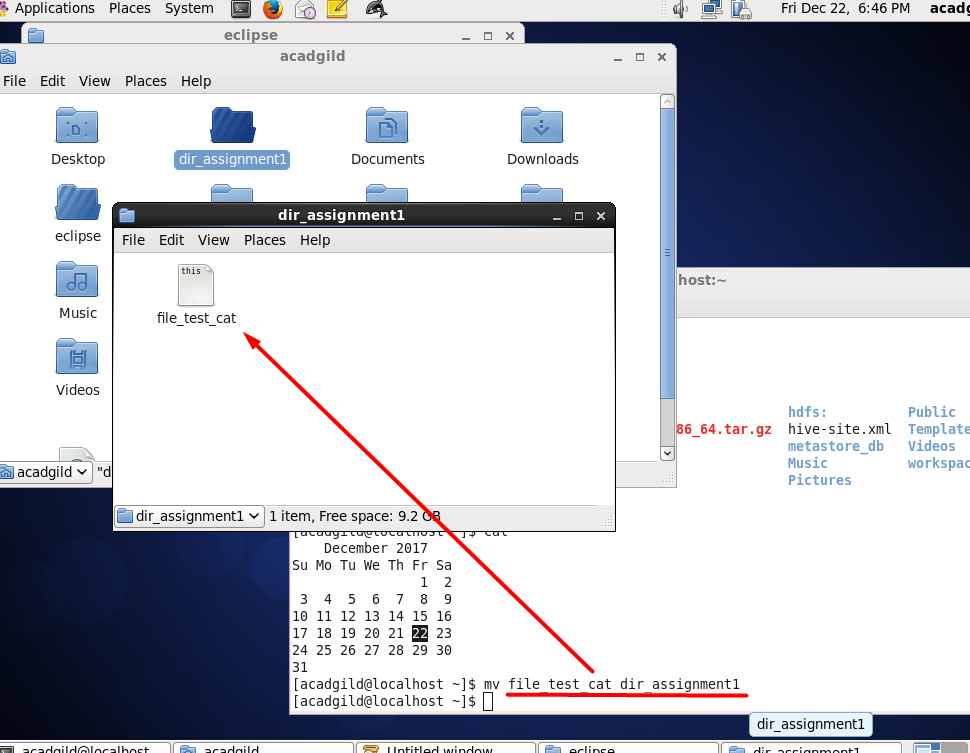
<https://www.computerhope.com/unix/udate.htm>

12.cal -  To display a calendar.

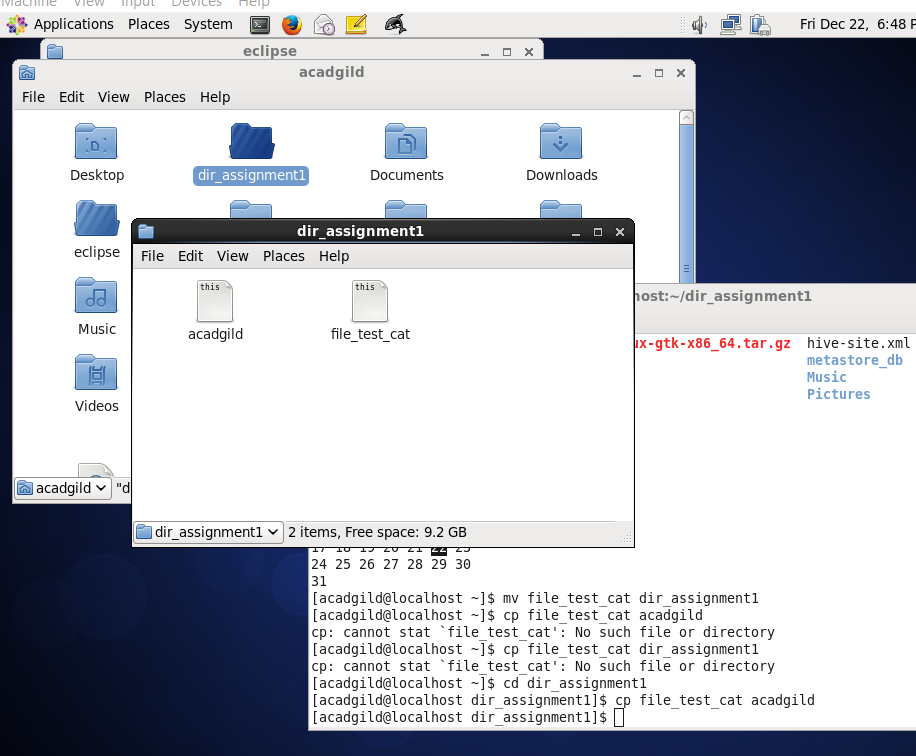
<https://www.tutorialspoint.com/unix_commands/cal.htm>



* 13.mv - **mv myfile yourfile :-**Move the file from "myfile" to "yourfile". This effectively changes the name of "myfile" to "yourfile".



* 14.cp - **cp myfile yourfile :-**This command copies the file "myfile" to the file "yourfile" in the current working directory. This command will create the file "yourfile" if it doesn't exist. It will normally overwrite it without warning if it exists. In order to get a prompt before overwritting, add a "-i" option like this to your command - **cp -i myfile yourfile.**Also, in order to copy a file from some location to your current directory you can do this - **cp -i /some/other/location .**
* **cp -dpr srcdir destdir :-**Copy all files from the directory "srcdir" to the directory "destdir" preserving links (-p option), file attributes (-p option), and copy recursively (-r option). With these options, a directory and all it contents can be copied to another directory.



15.which -  It allows user to pass several command names as arguments to get their paths in the system. “which” commands searches the path of executable in system paths set in $PATH [environment variable](http://www.thegeekstuff.com/2012/07/linux-export-command-examples/).

<http://www.thegeekstuff.com/2013/04/linux-which-whatis-whereis/>

