Lab 1

Hands-on Project 1.

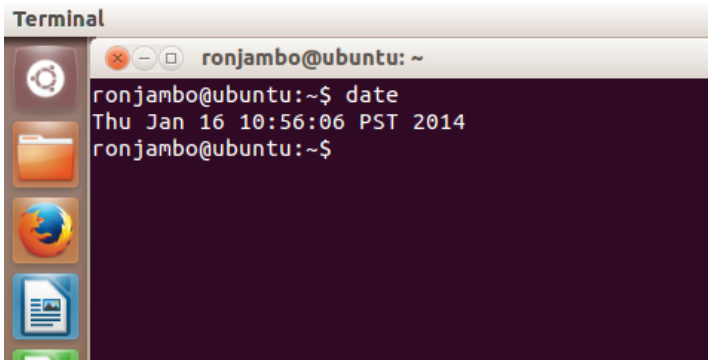
To display your system date:

1. Type **`date`** in the command line, and press **Enter**.

   A date similar to the following appears:

   ```
   Sat Nov 21 21:30:09 EST 2009
   ```

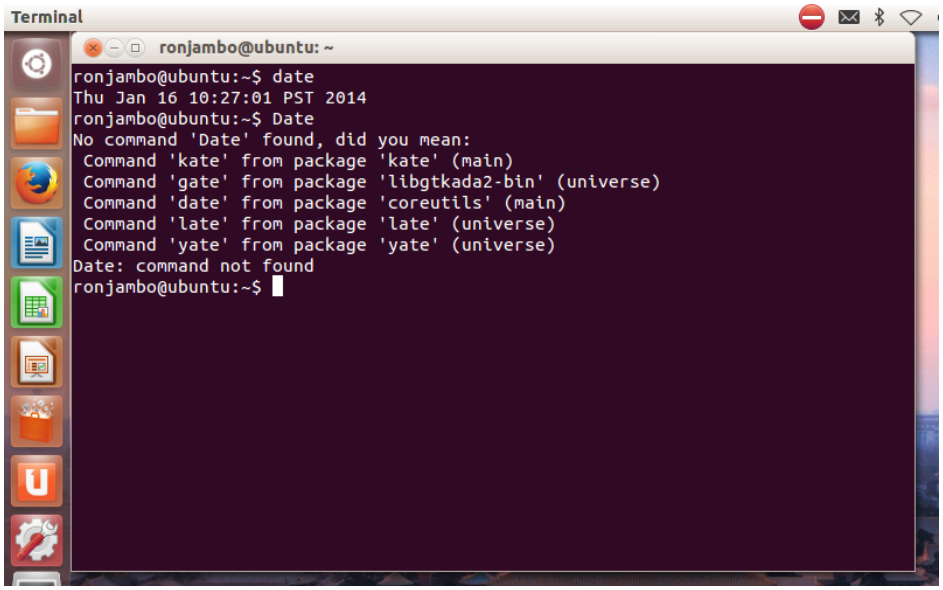
   You might see the abbreviation EDT (Eastern Daylight Time) instead of EST (Eastern Standard Time), or another time zone abbreviation, such as PDT (Pacific Daylight Time) or CST (Central Standard Time). Notice also that UNIX/Linux use a 24-hour clock.



2. Type **`Date`** in the command line, and press **Enter**. You see the following system error message:

   ```
   bash: Date: command not found
   ```

   The system error message appears because you must enter the date command, like most UNIX/Linux commands, in lowercase letters.



Hands-on Project 2

In this project, you use the **`cal`** command to display the current calendar, a Julian date calendar, and the historical calendar for July 1776.

To use the `cal` command:

1. Type **`cal`** in the command line, and press Enter. What calendar do you see? The calendar for the current month and year is displayed.
2. Type **`cal -j 2009`** in the command line, and press Enter. What type of calendar appears? A calendar appears that numbers the days of the year 1 – 365 instead of separating the days by the month of the year.
3. To determine the day of the week when the Declaration of Independence was signed, type **`cal 7 1776`** in the command line, and press Enter. In this case, the month and year are the command arguments. The Declaration of Independence was signed on a Thursday.

Hands-on Project 3

Use the **`clear`** command anytime you want a clean slate. This project enables you to clear the screen now.

To clear the screen:

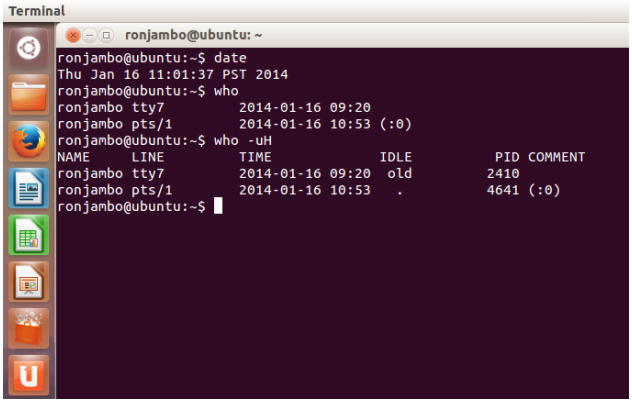
Type **`clear`** on the command line, and press Enter. The command prompt is now in the upper-left corner of your screen.

Hands-on Project 4

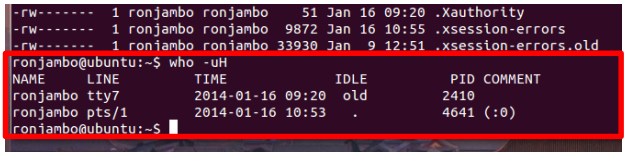
In this project, you use the command history capability of the Bash shell to recall commands you have used earlier. As you'll discover the more you use Linux, this command-line capability saves lots of time otherwise spent on repeated typing.

To use the command-line history:

1. Type **`date`** and press Enter.
2. Type **`who`** and press Enter.
3. Type **`who -uH`** and press Enter.



4. Type **`clear`** and press Enter.
5. Press the **`up arrow`** key four times. The **`date`** command is recalled to the command line. Do not press Enter.
6. Press the **`down arrow`** key twice. What command do you see? Press Enter to execute the command. I see the 'who –uH' command.

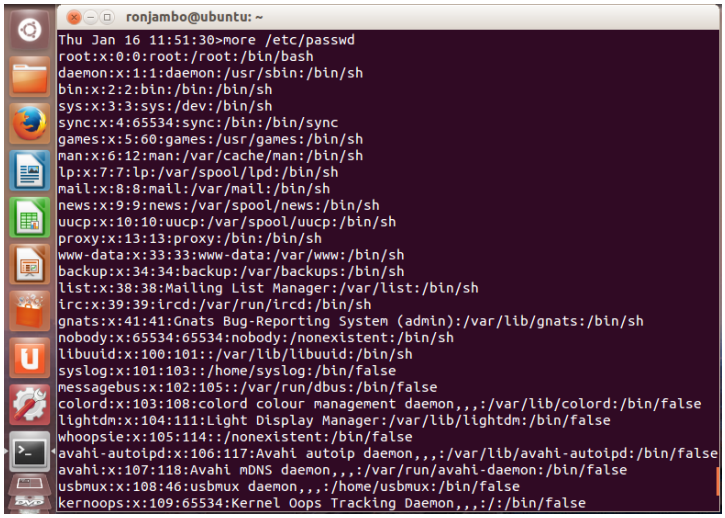


Hands-on Project 5

You can use the **`more`** and **`less`** commands to read a large file, screen by screen.

To view the contents of large files on the screen with the `more` command:

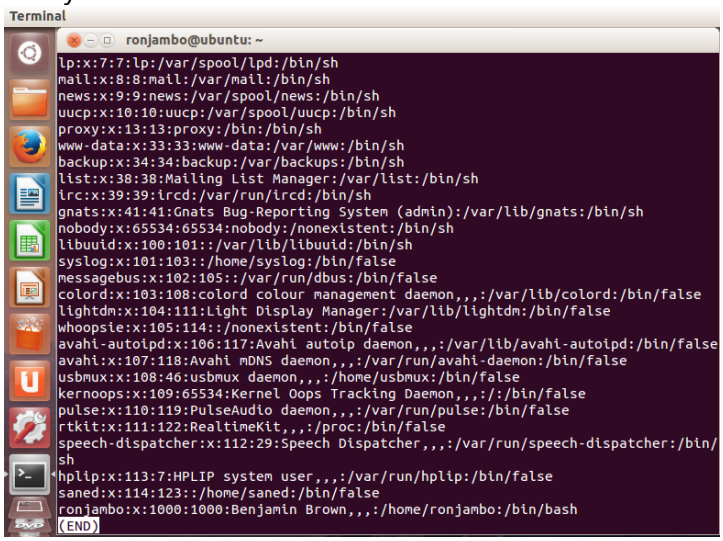
1. Type **`more /etc/passwd`** after the command prompt, and press Enter
2. Press **`Spacebar`** to scroll to the next screen.



3. Terminate the display by typing **q** (for quit)**.**

To view the contents of large files on the screen with the `less` command:

1. Type **`less /etc/passwd`** after the command prompt, and press Enter. You see a long file of text on your screen.



2. Press the **`down arrow`** key several times to scroll forward in the file one line at a time.
3. Press the **`up arrow`** key several times to scroll backward in the file one line at a time.
4. Press **`Pg Dn (or Page Down), Spacebar, z`**, or **`f`** to scroll forward one screen.
5. Press **`Pg Up (or Page Up)`** or **`b`** to return to a previous screen.
6. Terminate the display by typing **`q`** (for quit).

Hands-on Project 6

In this project, you will use the **`pwd`** command to view your working directory.

To display your current path:

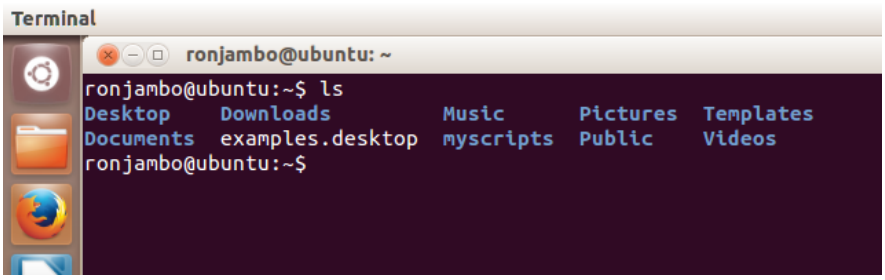
1. Type **`pwd`** and press Enter.
2. What is your current directory path? /home/ronjambo

Hands-on Project 7

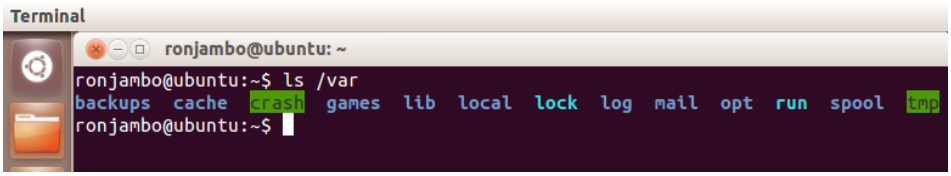
The **ls** command is one of the most useful commands. In this project, you will start by using **ls** to view your working directory. Next, you use **ls** with an argument to view a file and then a directory. For a more complete listing of information about the contents of a directory, you use the **-l** option, and finally you use the **-a** option to include hidden files in a directory listing.

To see a list of files and directories in your current working directory:

Type **ls** and press Enter. You see a list of file and directory names.

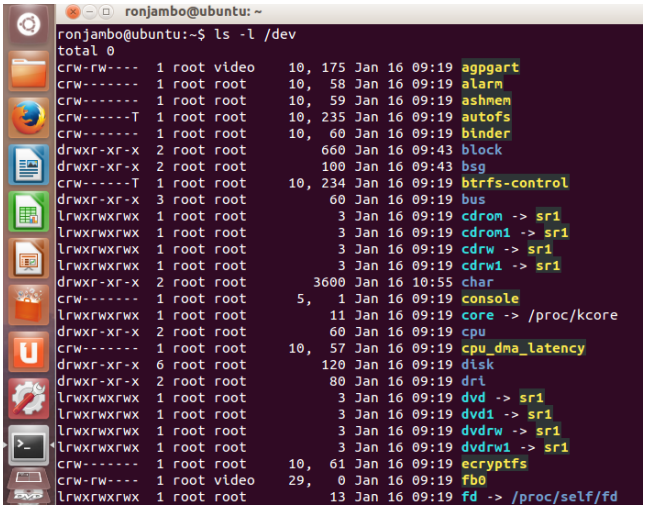


To see the contents of a directory other than your current working directory, give the directory name as an option to the **ls** command. For example, to see the contents of the **/var** directory, type **ls /var** and press Enter.

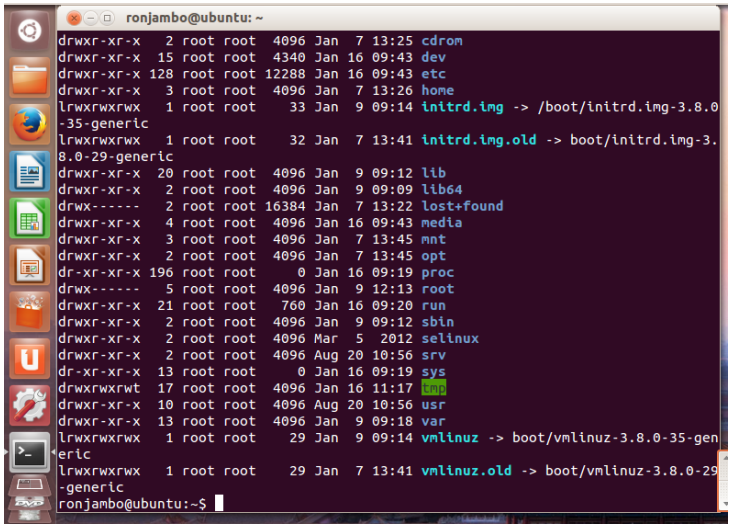


To use the `ls` command with the `-l` option:

1. Type **ls -l /dev** and press Enter. This shows a listing of block special and character special files in the **/dev** directory.

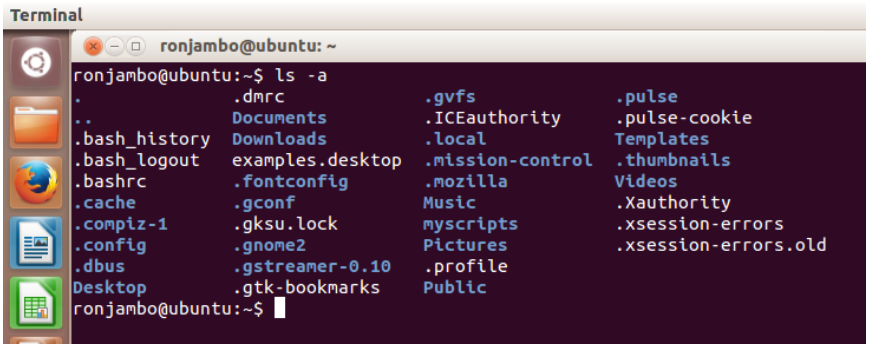


2. Type `ls -l /` and press Enter to view the contents of the root file system directory.



To list hidden files in your home directory:

1. Type **`clear`** and press Enter to clear the screen.
2. Type **`ls -a`** after the command prompt and press Enter.



Hands-on Project 8 (Optional. This is outside the scope of this class, but do it for fun anyways!!!)

Now onto something fun! Changing the prompt!

The **`PS1`** variable contains the configuration parameters for how your command-prompt line appears. In this project, you will view the contents of the **`PS1`** variable and then you configure the **`PS1`** variable. You should be using the default Bash shell and be logged in using your own account and home directory.

To view the `PS1` variable's contents and then to configure the variable:

1. Type echo $**PS1** and press Enter.
2. You see the contents of the **PS1** variable appear as:
   `[\u@\h \W]\$`
3. To change your prompt to display the date and time, type **PS1='\d \t>'** and press Enter. Type the command with no spaces between the characters, other than one space between **\d** and **\t**. Your prompt now looks similar to:

   Tue Jul 5 09:18:33>

4. To change your prompt to display the current working directory, type **PS1='\w>'** and press Enter. Your prompt now looks similar to:

   ~>

   The **\w** formatting character displays the **~** to represent the user's home directory.

5. To change your prompt to display the full path of the current working directory, you must use another environment variable, **PWD**. The **PWD** variable contains the full pathname of the current working directory. To display the **PWD** variable in the prompt, type **PS1='$PWD>'** and press Enter. (Notice that you must place the **$** in front of the environment variable name to extract its contents.) Your prompt now looks similar to:

   /home/jean>

6. If you are using a terminal window, close and open a new terminal window session, or log out and log back in and then access the command line. How does your prompt change from what you saw in Step 5? The prompt changes to appear more like an e-mail address showing the current user's name and the Linux distro being used.