Casey Bladow
CSIS 360 - Spring 2015
Assignment 11 - 25 points
Due Tuesday, April 28

Follow the **Creating Assignments** handout while performing the below steps. The purpose of this assignment is to learn about the . (dot) files, aliases, and shell functions. You will also learn about various administration tasks as described in Chapter 16 (Administration Tasks).

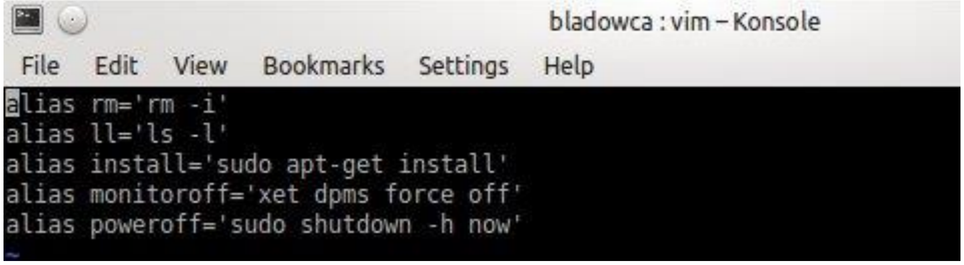
If any of you find that I am misusing or have incorrect terminology in any of my descriptions, please drop me a note with an explanation. I'm learning too!

1. This activity will make sure you have an understanding of . (dot) files and how they are executed when you login and you bring up a new shell. This was discussed in chapter 9.
   - Re-read page 294 in chapter 9. This important page tells you the order of files that are executed when you login and when you create a new shell. For example, the file *.profile* is executed once when you login and the file *.bashrc* executes for every new shell. This is why if you want to add to the PATH environment variable, you should do it in *.profile* and not *.bashrc* (which would append to it every time you opened a new shell).
   - Open a terminal window.
   - Type the command *less .bashrc* and look at it's contents. Hitting the space bar takes you down a page at a time. When you reach the bottom, you'll see the following towards the top of the screen.

   -
     ```
     if [ -f ~/.bash_aliases ]; then
         . ~/.bash_aliases
     fi
     ```

     What this does is that if the file *.bash_aliases* exists, execute it. This provides a convenient place for you to add aliases that you create. If this *if* statement is missing from your *.bashrc* file (it was there for me), add it.

   - Using the *vim* editor, create the file *.bash_aliases* (or edit it if it exists) and add some aliases to it. Aliases were discussed in chapter 9, and you should have tried a number of them in the previous assignment.
   - Be sure to test your aliases. Note, you will need to open a new shell (or type *bash*) for the aliases to be read and ready to use.
   - Open up the file *.bash_aliases* in the *vim* editor and ***do a window grab of the terminal window and include it in your assignment 11 document.***



```
alias rm='rm -i'
alias ll='ls -l'
alias install='sudo apt-get install'
alias monitoroff='xet dpms force off'
alias poweroff='sudo shutdown -h now'
~
```

2. This activity will make sure you have an understanding of creating your own functions. Some things cannot be done creating aliases.
   - Read about functions in chapter 9 (starting at the bottom of page 349-351). I had you skip this section in the previous assignment.
   - Open a terminal window and resize it vertically so that it has maximum height from the top to bottom of the screen (more room for me to see your results when you do the window grab).
   - Using the *vim* editor, edit the file *.bashrc* and duplicate the *if* statement for *.bash_aliases* and make it for *.bash_functions*. Hints (reminder) for the *vim* editor: go to the line with the *if*, type *3yy*(yank 3 lines) then press *P* (paste before, lower case p would paste after which would mix up the pasting). *x* will delete a char at the cursor.
   - create the file *.bash_functions* using the *vim* editor that contains the following functions. I'll be glad to help you with any of these functions in class if you need it.
     - tricopy - copies a file to 2 others

       ```
       tricopy a b c
       ```

       will copy file *a* to both *b* and *c*.

     - makedir - makes a directory and changes to it

       ```
       makedir exampleDir
       ```

       will create the directory *exampleDir* and change to it.

     - peek - list the table of contents of an archive (refer to chapter 5) and pipe it to the *less* utility.
     - `peek exampleDir.tar`
     - `peek exampleDir.tar.gz`
       `peek exampleDir.tar.bz2`

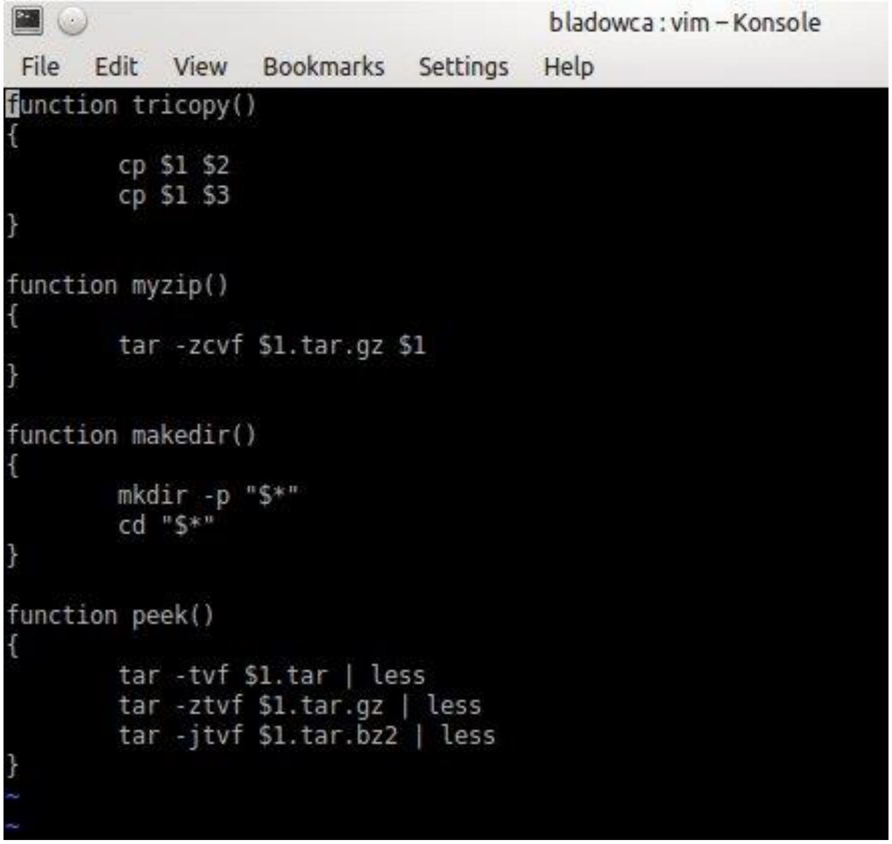
will list the table of contents for any of the above using the *less* utility. Note, there is nothing special that needs to be done if the file is compressed.

- myzip - will tar and compress (using gzip) a directory (refer to chapter 5)

  myzip exampleDir

  will tar and compress the directory *exampleDir* to the file *exampleDir.tar.gz*

- Be sure to test your functions. Note, you will need to open a new shell (or type *bash*) for the functions to be read and ready to use.
- Open up the file *.bash_functions* in the *vim* editor and ***do a window grab of the terminal window and include it in your assignment 11 document.***



```
function tricopy()
{
        cp $1 $2
        cp $1 $3
}

function myzip()
{
        tar -zcvf $1.tar.gz $1
}

function makedir()
{
        mkdir -p "$*"
        cd "$*"
}

function peek()
{
        tar -tvf $1.tar | less
        tar -ztvf $1.tar.gz | less
        tar -jtvf $1.tar.bz2 | less
}
~
~
```

3. Read chapter 16 (Administrative Tasks) while at the keyboard experimenting with the commands as you read the chapter. Notes to keep in mind as you read the chapter.
    - You have already read the beginning of the chapter (pages 594-598) on configuring user and group accounts.
    - You should read the section on backing up files (pages 599-605) but you don't need to experiment with the commands (unless you really want to). You should already be familiar with the *tar*utility from chapter 5.
    - You should read the section on partitioning a hard disk (pages 611-614) but I wouldn't be messing around too much.
    - You've already installed MySQL in a previous assignment and did a number of things described so you can skip the MySQL section (pages 628-635). Go ahead and read through it if you want more information on how to use it.
    - ***Answer the following questions in your assignment 11 document.***
        - ***What utility will list the most CPU-intensive processes?*** top
        - ***What command will list all the running processes? Be sure to give the necessary options with it.*** ps aux | less
        - ***What utility will run a process once at a later time?*** at
        - ***What utility generates virtual memory information along with (limited) disk and CPU activity data?*** vmstat
        - ***What are the backup utilities discussed in the chapter to construct full or partial backups of a system?*** tar, cpio, dump/restore
        - ***What partition editor utility reports on and manipulates hard disk partitions?*** parted
        - ***What utility can be used to communicate with a user who is logged in on the local system?*** write
        - ***What utility effectively communicates immediately will all users who are logged in?*** write
        - ***What utility will kill a running process?*** -KILL
        - ***What utility will display the names of open files?*** lsof

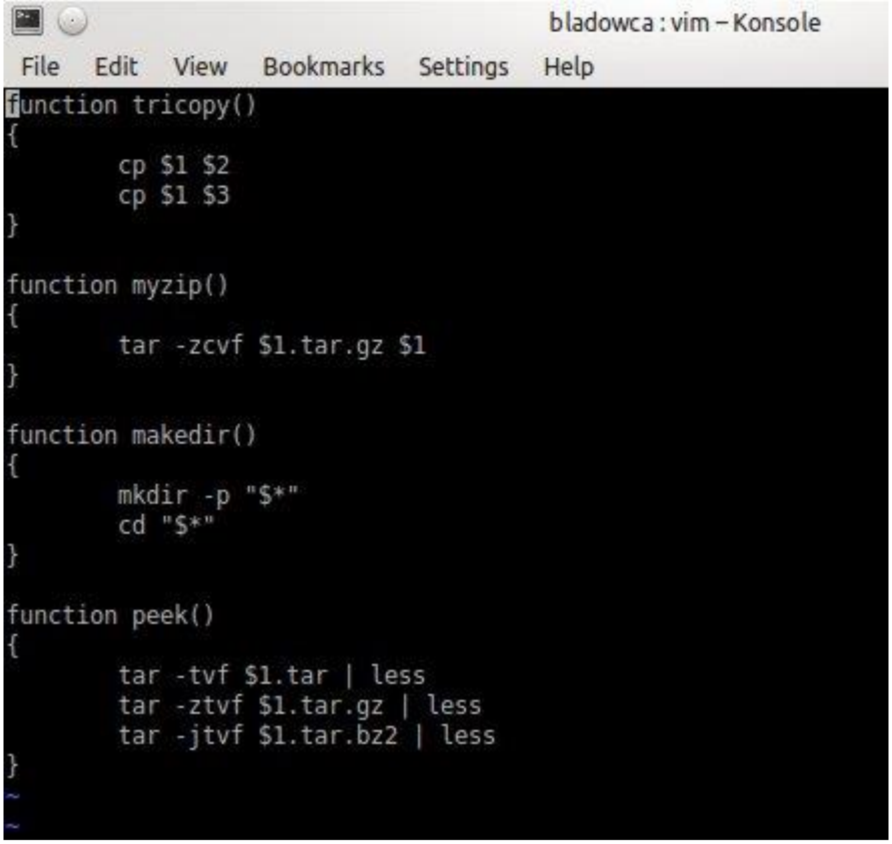
4. You just read about *cron* and *anacron* in chapter 16. This activity will give you a better understanding of them and have you demonstrate their usage.
    - Read the following two web articles, the first one is on *cron* and the other on *anacron*. This will help give you a better understanding of them and prepare you for this activity.

  - **http://www.thegeekstuff.com/2009/06/15-practical-crontab-examples**
  - **http://www.thegeekstuff.com/2011/05/anacron-examples**

- ***In your assignment 11 document, describe the difference between cron and anacron. Include when each would / should be used.***

  Cron can be scheduled by any normal user, Anacron can only be used by a super user. Cron expects the system to be running 24x7, Anacron doesn't expect that. Cron is ideal for servers, Anacron is ideal for desktops and laptops. Cron should be used when a job has to be executed at a particular hour and minute. Anacron should be used when a job has to be executed irrespective of hour and minute.

- Perform the following for *cron* jobs.
  - Create a cron job that will tar and compress your *assign9* directory to the file *assign9.tar.gz* in your home directory. Make the command execute every minute. You should be able to see this in action. If you delete *assign9.tar.gz*, it should be created again within a minute.
  - Create a cron job that will tar and compress the *ubuntuDir* directory in the *assign9* directory to the file *ubuntuDir.tar.gz* in your home directory. Make the command execute at 3:00AM every Wednesday. You likely won't see the results of this.
  - Create a cron job that will tar and compress the *smaugDir* directory in the *assign9* directory to the file *smaugDir.tar.gz* in your home directory. Make the command execute at 2:00AM on the first day of every month. You likely won't see the results of this.
  - Give the command
    crontab -l
    then ***do a window grab of the terminal window and include it in your assignment 11 document***

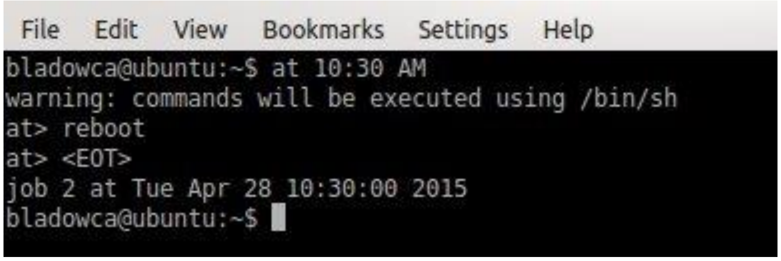


```
function tricopy()
{
        cp $1 $2
        cp $1 $3
}

function myzip()
{
        tar -zcvf $1.tar.gz $1
}

function makedir()
{
        mkdir -p "$*"
        cd "$*"
}

function peek()
{
        tar -tvf $1.tar | less
        tar -ztvf $1.tar.gz | less
        tar -jtvf $1.tar.bz2 | less
}
~
~
```

- After the screen grab, you're welcome to delete the crontab entries.

- Perform the following for *anacron* jobs.
    - Create 3 anacron jobs that will create the same three .tar.gz files as described in the above *cron* jobs example. Name the entries *mybackup1*, *mybackup2*, and *mybackup3*. Make the commands execute every day with a 1 minute delay (so you don't have to wait around for it). Once you've made the entries in */etc/anacrontab* you will need to reboot your computer (as you should know from reading about anacron). Reboot your computer at the time you specify using the *at* utility that you learned about in chapter 16. I would suggest just a couple minutes ahead of time so you don't have to wait long. ***Before the computer reboots, do a window grab of the terminal window showing the 'at' command to reboot and include it in your assignment 11***

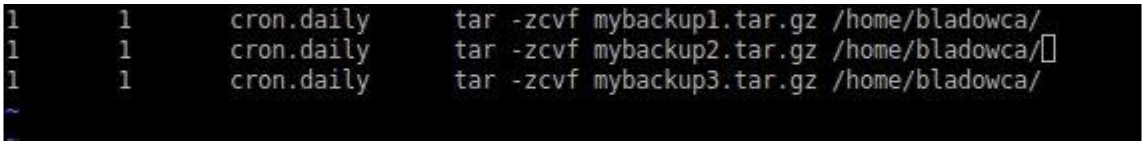
document.



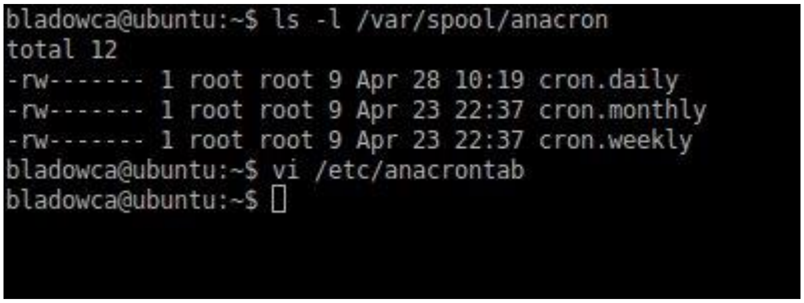
```
bladowca@ubuntu:~$ at 10:30 AM
warning: commands will be executed using /bin/sh
at> reboot
at> <EOT>
job 2 at Tue Apr 28 10:30:00 2015
bladowca@ubuntu:~$
```

You may need to install the *at* utility if you did not do so when reading chapter 16. Be sure you give yourself enough time to do the window grab and save it before the computer reboots! Also be sure that you don't have any open applications that you haven't saved.

- After the reboot, you should see the files *assign9.tar.gz*, *smaugDir.tar.gz*, and *ubuntuDir.tar.gz* in your home directory after the one minute delay specified in the *anacron* entry.
- From the terminal window, type the commands
  cat /etc/anacrontab
  ls -l /var/spool/anacron
  and ***do a window grab of the terminal window and include it in your assignment 11 document.***



```
1       1       cron.daily      tar -zcvf mybackup1.tar.gz /home/bladowca/
1       1       cron.daily      tar -zcvf mybackup2.tar.gz /home/bladowca/
1       1       cron.daily      tar -zcvf mybackup3.tar.gz /home/bladowca/
~
```



```
bladowca@ubuntu:~$ ls -l /var/spool/anacron
total 12
-rw------- 1 root root 9 Apr 28 10:19 cron.daily
-rw------- 1 root root 9 Apr 23 22:37 cron.monthly
-rw------- 1 root root 9 Apr 23 22:37 cron.weekly
bladowca@ubuntu:~$ vi /etc/anacrontab
bladowca@ubuntu:~$
```

- After the screen grab, you're welcome to delete the entries in */etc/anacrontab*.

5. ***What did you think of this assignment? How long did it take you? Do you have any suggestions for the next time I teach this class?*** Long and

frustrating. Right around 6 hours. I had issues with my bash and anacrontab not working right.

6. ***What did you learn from this assignment?***

   Cron is a lot easier to use than anacron. AND there’s a GUI for it!