**Regis University CC&IS**

**CS465 Unix**

**Lab Homework #4**

**Matt Hartigan**

**2-Apr-2019**

**\*answers below in red\***

Write the following four Bourne shell scripts, and submit listings of the contents of each.

Do ***not*** just include a screen shot – include the actual ***text*** commands in each script file.

Points that will be given for correct implementation of the items listed within each question.

### See the end of the document for submission instructions.

1. (14 points total, allocation noted below)

Write a Bourne shell script called **hw1**, which gets executed when the user logs in.

a) (12 points) The script should display the message “Good Morning” / “Good Afternoon” / “Good Evening” / "Good Night" depending upon the time at which the user logs in. Use the following hours to detemine which message to display:

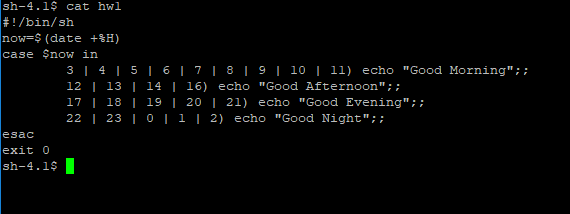
For hours 3 – 11: Morning

For hours 12 – 16: Afternoon

For hours 17 – 21: Evening

For hours 22 – 2: Night

Place the contents of the script file here:



b) (2 points) How can you make this script run every time the user logs in?

You can append the above script to one of the files that is executed every time you log in. For example, you can add it to .profile, or /etc/profile

Just don’t forget to remove the ‘exit 0’ command at the end, otherwise the login shell will terminate automatically! Also, it is not necessary to include the #!/bin/sh if appending the full script contents to .profile.

A simpler way to accomplish the same task is to put the absolute path name to the hw1 script in your.profile file. In this case you can add the following to .profile:

/home/harti149/hw1

2. (23 points total, allocation noted below)

Write a Bourne shell script called **hw2** which will:

Move files to immediate subdirectories. The command line arguments will be PAIRS of files and subdirectories. The script will move each file to the named subdirectory, if the subdirectory exists.

Example:

$ **hw2 file1.c cprogs file2 memos**

The above command would do the following, if both subdirectories exist:

Move **file1.c** to immediate subdirectory **cprogs**

Move **file2** to immediate subdirectory **memos**

**Implementation:** You must implement this script using a **while** loop that contains **shift** commands to access the filename and subdirectory arguments (NOTE: **$#** will change every time you use the **shift** command).

(7 pts) The script should first verify that an EVEN number of arguments (2 or more) were entered. If not, use a **HERE document** to display an error message and the correct argument format, and then exit the script with exit code 1.

Then, before moving each file, the script should verify that the both the named file and the named subdirectory exist.

- (4 points) If the file does not exist, display an error message which includes the name of the file that does not exist.

- (4 points) If the subdirectory does not exist, display a message stating which subdirectory does not exist and saying which file was not moved.

- (4 points) If both the file and the subdirectory exist, move the file and display a message confirming that the file was moved and to where.

(4 points) Shift and loop until all file/subdirectory pairs have been processed.

Sample Runs:

$ **hw1 myfile**

Error: Must have an even number of arguments (2 minimum)

Usage: hw1 file subdir [[file subdir]...]

$ **hw1 file1.c cprogs file2 memos file3 pending**

File file1.c moved to subdirectory ./cprogs

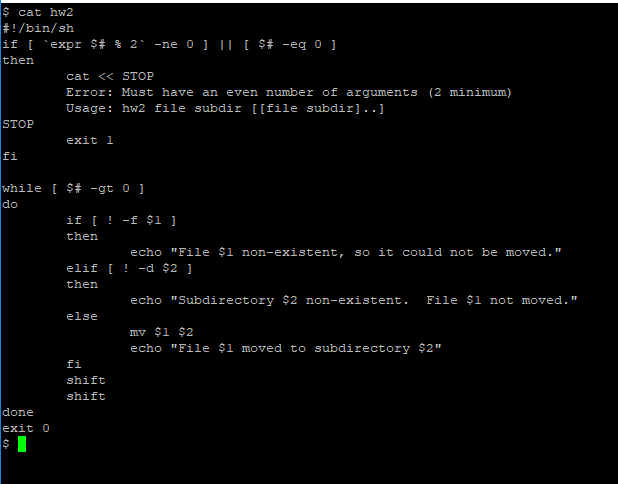
subdirectory ./memos non-existent. File file2 not moved.

File file3 non-existent, so it could not be moved.

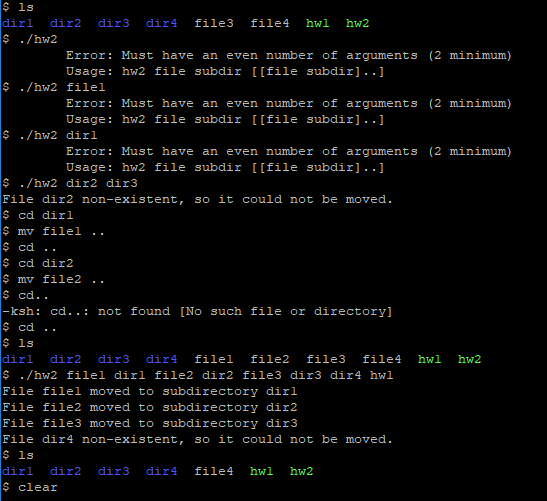
$

Place the contents of the script file here:

hw2 script file



Example of checks performed:



3. (43 points total, allocation noted below)

Write a Bourne shell script called **hw3**, which will:

Take three **command line arguments**, a 3-letter month string (eg. Sep), a day number (eg. 18), and a 4-digit year (eg. 1996). No dates before Jan 1, 1940 or after the current year should be accepted.

(6 points) First, the script should validate that 3 arguments were entered. If not, the script should display an error message that contains the correct argument format, and then exit with **exit code 3**.

Second, the script should error check that valid values for the month, day, and year have been entered.

(7 points) **For the year**: No year before1940 or after the current year should be accepted. If year is invalid, issue an error message and exit the script with **exit code 13**.

*NOTE: The current year should be accessed via the system* ***date****, not hardcoded into the script.*

*The* ***date*** *command should not be used anywhere else in this script.*

(6 points) **If a valid year was entered**, display whether the year entered was a leap year (Note: For dates in this range, you will only need to check if a year is evenly divisible by 4, to determine if it is a leap year). Save the leap year status.

NOTE: This should be displayed as long as a valid year is entered -- even if there are errors found later in the month or day.

(6 points) **For the month** (without using the **date** command): The month must be a valid three letters abbreviation and begin with a capital letter (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, or Dec). If month is valid, convert it to a month number (1-12). If month is invalid, issue an error message and exit the script with **exit code 11**. *Hint: Use a* ***case*** *statement.*

(10 points) **For the day number** (without using the **date** command): After verifying a valid month and year has been entered, verify that the number of days entered is possible for that particular month (note: If the month entered is Feb, you will need to determine if the year is a leap year – could be saved from check above). If day number is invalid, issue an error message and exit the script with **exit code 12**.

(7 points) After verifying the correctness of all arguments, display the date entered in the format:

**MM/DD/YY**

Note: Leading zeros do NOT have to be displayed for month or day,

but the year should always be displayed with 2 digits.

(1 point) Exit with **code 0** if the script runs successfully.

Sample Runs:

$ **hw3 Nov 3 1955**

1955 was not a leap year

Date entered was 11/3/55

$

$ **hw3 Apr 31 1940**

Invalid day entered for Apr

$

As a side note, I think the “1940 was a leap year.” got left out of this output. 1940 was a leap year according to this source (<https://kalender-365.de/leap-years.php>)

$ **hw3 Apr 12 2004**

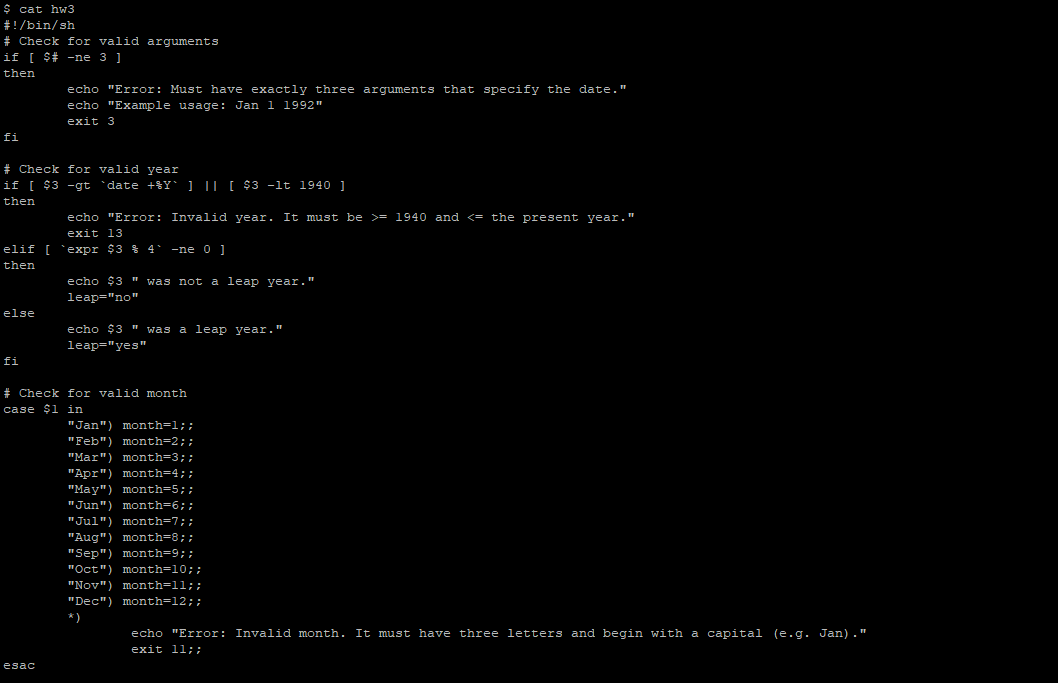
2004 was a leap year

Date entered was 4/12/04

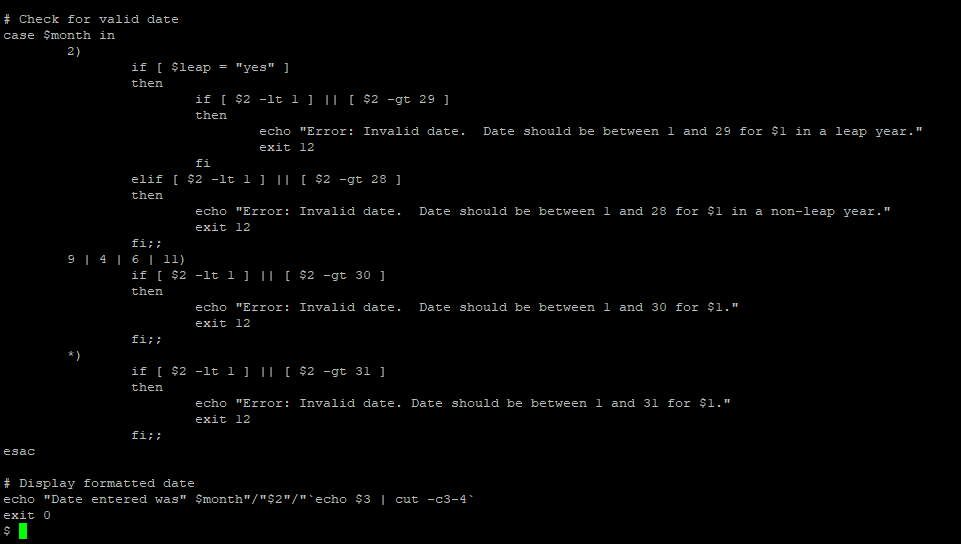
$

Place the contents of the script file here:

hw3 script file (2 snips)



hw3 script file continued



4. (20 points total, allocation noted below)

Write a Bourne shell script called **hw4**, which will:

Take the exit code from **hw3** as an argument.

(4 points) Verify an argument was entered.

If no argument was entered, issue an error message explaining what argument is necessary and then exit the script with exit code 1.

Determine if **hw3** ran successfully.

(4 points) If it did, say so.

(12 points) If it did not, say so, and evaluate the exit code (3, 11, 12, or 13) to display the reason it failed.

Exit with exit code 0.

Sample runs:

$ **hw4**

Missing argument for hw4

Usage: hw4 $?

$

$ **hw3 Nov 3 1955**

1955 was not a leap year

Date entered was 11/3/55

$ **hw4 $?**

hw3 script ran successfully

$

$ **hw3 Nov 3**

Argument(s) missing.

Usage: hw2 MMM DD YYYY

$ **hw4 $?**

hw3 script did not run successfully.

Invalid number of arguments.  
 $

$ **hw3 Nov 3 1939**

Year must be between 1940 and now.

$ **hw4 $?**

hw3 script did not run successfully.

Invalid year entered.

$

$ **hw3 Nob 3 1955**

Nob not valid month

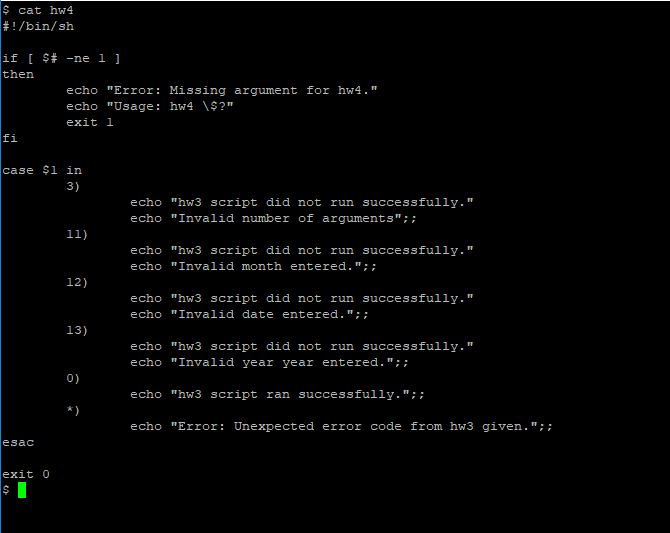
$ **hw4 $?**

hw3 script did not run successfully.

Invalid month entered.

$

Place the contents of the script file here:



hw4 script file

# Submission

This homework assignment is due by midnight Sunday (last day of Week 4).

Submit a filled in copy of this Word document to the **Homework Assn 4** drop box (located under the Dropbox tab in the online course).

Before submitting the Word file with your answers, you MUST rename it as follows:

### Lastname-hwk4.docx

For example:

### Smith-hwk4.docx