## Exercise 1 – Create a directory structure

|  |
| --- |
| **Task** |
| Create the following directory structure within your **home** directory.  You can perform this task using only **mkdir** and **touch**, and **cd**.  To check your structure, use **ls** and **tree**. |
| ~/  |  |  films/  | |  --------------------------------------------  | | |  | | |  horror/ comedy/ action/  theHowling bigMommasHouse rambo  | ghostbusters   -------------------- ghostbusters2  | | shawnOfTheDead  | |  zombie/ slasher/  dawnOfTheDead halloween  fridayThe13th  predator |

## Exercise 2 – Change a directory structure

|  |
| --- |
| **Task** |
| The following tasks involve: **mv, rm, glob matching, and brace expansion**  Your films folder should currently look like this: |
| ~/  |  |  films/  |  |  --------------------------------------------  | | |  | | |  horror/ comedy/ action/  theHowling bigMommasHouse rambo  | ghostbusters   -------------------- ghostbusters2  | | shawnOfTheDead  | |  zombie/ slasher/  dawnOfTheDead halloween  fridayThe13th  predator |

Perform the following actions

1. Navigate to the 'zombie' folder. Stay in the zombie folder throughout this exercise.
2. Move 'predator' from the 'slasher' folder to the 'action' folder.
3. Rename 'rambo' to 'firstBlood'.
4. Create a sub-folder within comedy called 'supernaturalComedy'.
5. Using only one command, move both 'Ghostbusters' films into the new folder.
6. Delete 'bigMommasHouse' in interactive mode.
7. With a single command, create ‘rockyI’, ‘rockyII’, … ‘rockyV’ in the action folder.

# Exercise 3 – The vi Editor

Use vi to create the ‘accounts’ file:

**PIN534:Sheldon Cooper:1:1024**

**PIN756:Leonard Hofstadter:2:620**

**PIN769:Howard Wolowitz:4:213**

**PIN210:Rajesh Koothrappali:3:556**

The first field is the pin. The second field is the account holder’s name. The third field is the account number. The fourth field is the account's balance.

# Execise 4 - Data Commands

Commands Needed: **wc, fgrep, cut, sort**

1. Count the number of characters in the ‘accounts’ file.
2. Count the number of words in the ‘accounts’ file.
3. Count the number of lines in the ‘accounts’ file.
4. Display the line containing ‘Sheldon Cooper’.
5. Display the account that has id of 3. (tip: look for ":3:")
6. Display the account with a balance of 620.
7. Retrieve and display the name of every person represented in the ‘accounts’ file.
8. Retrieve the pin numbers for every person represented in the ‘accounts’ file.
9. Retrieve the balance and the real name for every person in the ‘accounts’ file.
10. Display the entries in the ‘accounts’ file sorted by account balance in descending order.

# Exercise 5a: Chmod

Create a regular file and run through the following steps:

1. Alter permissions using **chmod** so that you have full access to the file and nobody else has any access. Verify that this has worked.
2. Give yourself read-write access, the file's group read access only and no access to anyone else.
3. Give yourself read-execute access, and execute only access to all others.

Create a directory.

1. Give yourself read and execute access but no write access. What does this prevent you from doing? Creating
2. Give yourself all privileges except for read access. What does this prevent you from doing?listing directory contents
3. Deny yourself execute access on the directory. What does this prevent you from doing?

# Exercise 5b: Using the Find Command

From the films directory:

1. Locate all files named ‘shawnOfTheDead’ in your films directory and sub-directories.
2. Locate all files in all subdirectories that were modified in the last 30 minutes.
3. Locate all files in all subdirectories with either ‘the’ or ‘The’ in the name.
4. Turn off write permission for all files that contain ‘the’ or ‘The’ in the name.
5. Interactively remove all files with ‘the’ or ‘The’ in the name. (You can keep the files if you answer ‘n’ when prompted.)

# 

# Exercise 6 – Stream Redirection on the Command Line

Perform the following actions using a single command on the command line.

1. Display the contents of the ‘accounts’ file in upper case.
2. Display the contents of the ‘accounts’ file with all the colons changed to spaces.
3. Retrieve the line containing ‘Sheldon Cooper’ from the accounts file and redirect it into a file called "sheldon". If the file "sheldon" already exists then overwrite the file.
4. Retrieve the line containing ‘Leonard’ and append it into the sheldon file.
5. Use echo and wc to count the characters in the word “characters”.
6. Display the number of lines in the ‘accounts’ file without displaying the name of the file.
7. Use sort to order the output of ls -l based on various different fields.

|  |
| --- |
| Exercise 7 – Regular Expressions Paste the following data into a file:  07999234123  075435345623  07AAAAAAAAA  07bbbbbbbbb  B41RTG  01223567345  A345GUF  B3GHJ  768745  KL563478K  kl563478kx  [l563478k  6535345354554234  B45JKHL  65-34-76  012234567867  07856456636  BB4JUK  56-567-67  B546HUY  BB345614H  07666345234  45-34-25  0122345675GREP  JH761423G  Use **grep or egrep** to extract the following:   1. Mobile phone numbers (11 digits starting in '07') 2. National insurance numbers (2 letters, 6 digits, 1 letter) 3. Cambridge telephone numbers (11 digits starting in '01223') 4. Bank sort codes (2 digits, dash, 2 digits, dash, 2 digits) 5. B-reg number plates ('B', 1-3 digits, 3 letters) 6. All lines that do not contain letters |

## Exercise 8 - $, $(( ... )), $[ ... ], $( ... )

Perform the following actions on the command line.

1. Set a variable x equal to "Andy". Set a variable y equal to "Bruce". Try the following: echo xy x y $x$y $x $y
2. Set the variable x equal to 4 and the variable y equal to 9. Display the the result of summing the contents of variable x and variable y.
3. Set the variable x equal to the output of the command: ls  
   Display the contents of variable x.
4. Retrieve all the balances from the accounts file and store the result in a variable called "balances". Display variable "balances".
5. Display the result of 4 + 3 \* 2. Display the result of (4+3)/2.
6. Create a file that has some mathematical formula. Send the contents of the file to bc (bench calculator) and store the result in a shell variable.
7. Use vi to create a file with “:” on line 1 and “6/3” on line 2. Using line 1 as the delimiter and line 2 as the field, cut and display the real names from the accounts file.

## Exercise 9 - Job Control

|  |
| --- |
| **Perform the following tasks** |
| 1. Run the sleep commands for 1000 seconds in the background |
| 1. Use the ps command to identify the PID of the sleep command |
| 1. Kill the sleep command and check the sleep process was terminated successfully. |
| 1. Start another sleep command for 1000 seconds in the background |
| 1. Use the jobs command to identify the JID of the sleep command |
| 1. Bring the sleep command into the foreground |
| 1. Kill the sleep command and check the process was terminated successfully. |