**Tutorial 2**

|  |  |
| --- | --- |
| mkdir | Create a new directory |
| ls -l | Display a detailed listing of a directory |
| ls -la | Display a detailed listing of a directory, including hidden files |
| cd | Change directory |
| cd ~ or cd | Change to home directory |
| cp -r | Copy a directory and its contents |
| mv | Move or rename a file or directory |
| rm -r | Remove a directory and its contents |
| rm -rf | Remove a directory and its contents without prompting |
| touch | Create an empty file |
| cat | Display the contents of a file |
| more | Display the contents of a file, one screen at a time |
| less | Display the contents of a file, allowing scrolling |
| sort | Sort the lines of a file |
| head -n | Display the first n lines of a file |
| tail -1 | Display the last line of a file |
| grep | Search for a pattern in a file |
| uniq | Display unique consecutive lines in a file |

**Tutorial 3**

| **Question** | **Command** |
| --- | --- |
| Create directory structure with relative pathnames | **mkdir final final/week1 final/week2 final/week3 final/week3/backup final/week3/finalback** |
| Create empty files with absolute pathnames | **touch /home/user/.answers.txt /home/user/questions.txt** |
| Display all filenames in directory | **ls -a final** |
| View contents of file with relative pathname | **cat backup/week2/.answers.txt** |
| Change to directory with absolute pathname | **cd /home/user/backup** |
| Verify current directory | **pwd** |
| Copy file with relative pathnames | **cp week2/questions.txt ../** |
| Delete file with relative-to-home pathname | **rm ~/backup/week2/questions.txt** |
| Safely remove directory and contents with absolute pathname | **rm -r /home/user/backup/week3** |
| Will command work to remove directory? | Answer with "yes" or "no" and reason |
| Remove files with relative-to-home pathname | **rm ~/final/\*.txt** |
| List files with absolute pathname | **ls /home/user/????** |
| List files with relative pathname | **ls \*[0-9]\*** |
| List files with relative-to-home pathname | **ls ~/[0-9]\*[^0-9]** |
| View contents of files with 5 consecutive numbers | **cat ?????** |
| Display message | **echo "\*\*\* Hello \*\*\*"** |
| Display message with quotation marks | **echo "\"This is my message\""** |

**Tutorial 4**

| **Question** | **Answer** |
| --- | --- |
| List the number of digits for the following numbering systems: | Decimal: 10, Binary: 2, Octal: 8, Hexadecimal: 16 |
| Write a simple chart to show which values are represented for letter A - F for a hexadecimal number. | A: 10, B: 11, C: 12, D: 13, E: 14, F: 15 |
| How many binary digits does 1 octal digit represent? | 3 |
| How many binary digits does 1 hexadecimal digit represent? | 4 |
| Use manual numbering conversion to complete the table displayed to the right. | N/A |
| Write the chmod command (using the symbolic method) to set “pass-through” permissions for your home directory using an absolute pathname. | chmod a=r+x ~ |
| Write a Linux command to verify that permissions where set. | ls -ld ~ |
| Perform a binary to octal numbering conversion for the permissions: r w x - - x - - x | 101 101 001 -> 551 |
| Write single Linux command to set “pass-through” permissions for your home directory, using the absolute method (i.e. octal numbers). | chmod 711 ~ |
| Write a single Linux command to add read permissions for same group members for the ~/tests directory. | chmod g+r ~/tests |
| Write a single Linux command to remove write permissions for same group members and other group members for the ~/projects directory. Use the symbolic method. | chmod g-w,o-w ~/projects |
| Write a single Linux command to set the permissions for the ~/assignments directory to the following using the absolute method (i.e. octal numbers): r w x r - x - - x. Show your work to perform a binary to octal conversion. | Binary: 101101001 -> Octal: 651 |
| Write the command below using octal numbers and using a relative-to-home pathname. | chmod 777 ./mydir |
| Assume that you just issued the command: chmod u=rwx,go=x ~/linux/content. What would be the new permissions for the “content” directory? | rwxr-xr-x |
| Assume that you just issued the commands: umask 077, mkdir mydir, touch mydir/myfile.txt. What would be the permissions for the newly created directory and regular file? (show your work) | Directory: 700 (777 - 077 = 700), Regular file: 600 (666 - 077 = 600) |

**Tutorial 5**

| **Question** | **Command** | **Options and Arguments** |
| --- | --- | --- |
| Write a single Linux command to provide a detailed listing of all files in the /bin directory, sending the output to a file. | **ls -la /bin > projects/listing.txt** | **-l** lists the files in a long format, **-a** shows hidden files, **>** redirects output to a file |
| Write a single Linux command to redirect the stderr from the command: cat a.txt b.txt c.txt to a file called error.txt. | **cat a.txt b.txt c.txt 2> assignments/error.txt** | **2>** redirects standard error output to a file |
| Write a single Linux command: cat ~/a.txt ~/b.txt ~/c.txt and redirect stdout to a file called “good.txt” and stderr to “bad.txt” | **cat ~/a.txt ~/b.txt ~/c.txt > tests/good.txt 2> tests/bad.txt** | **>** redirects standard output to a file, **2>** redirects standard error output to a file |
| Write a single Linux command to redirect the stdout from the command: cat a.txt b.txt c.txt to a file called wrong.txt. | **cat a.txt b.txt c.txt >> projects/wrong.txt 2> /dev/null** | **>>** appends standard output to a file, **2>** redirects standard error output to /dev/null (null device) |
| Write a single Linux pipeline command to display a detailed listing of the projects directory but pause one screen at a time. | `ls -laR ~/projects | less` |
| Write a single Linux pipeline command to display the sorted contents (in reverse alphabetical order) of the “linux” directory. | **ls -r linux/ | sort -r** | **-r** lists files in reverse order, **|** pipes the output to the **sort** command to sort in reverse order |
| Assume that the text file called “.answers.txt” contains 10 lines. Write a single Linux pipeline command to only displays lines 5 through 8 for this file. | **sed -n '5,8p' .answers.txt** | **-n** suppresses output, **5,8p** prints lines 5 through 8 |
| Write a single Linux pipeline command to only display the contents of the “assignments” directory whose filenames match the pattern “murray”. | **ls -l /home/user/assignments/\*murray\*** | **-l** lists the files in a long format, **\*** is a wildcard character to match any characters |
| Write a single Linux pipeline command to display the number of characters contained in the file called “.answers.txt”. | **wc -c ~/.answers.txt** | **-c** counts the number of characters in a file |
| Write a single Linux pipeline command to display the number of lines contained in the file called “questions.txt”. | **wc -l questions.txt** | **-l** counts the number of lines in a file |
| Write a single Linux pipeline command to display only the first 10 characters of each filename contained in your current directory. | `ls | cut -c1-10 |

**Tutorial 6**

| **Linux Command** | **Description** |
| --- | --- |
| **cp mytext.txt user1@tech.myserver.com:~/** | Copies the file **mytext.txt** from the current directory to the home directory of user **user1** on the remote server **tech.myserver.com** |
| **cp mytext.txt user1@tech.myserver.com:~/yourtext.txt** | Copies the file **mytext.txt** from the current directory to the home directory of user **user1** on the remote server **tech.myserver.com** and renames the file as **yourtext.txt** |
| **scp ~/project/linux.txt username@linux.techie.org:~/** | Copies the file **linux.txt** from the home directory of the local user to the home directory of the remote user **username** on the server **linux.techie.org** |
| **sftp saulm@tux.senecac.on.ca** | Connects to the remote server **tux.senecac.on.ca** with the username **saulm** for transferring files |
| **lpwd** | Displays the current working directory on the local server |
| **ls** | Views files on the local server |
| **ls remote\_directory** | Views files in the remote directory |
| **get answers.txt** | Downloads the file **answers.txt** from the current directory of the remote server |
| **put questions.txt ~/documents/tests/** | Uploads the file **questions.txt** from the local server to the directory **~/documents/tests/** on the remote server |
| **quit** or **exit** | Quits the current sftp session |
| **mail -s "Important Message" murray.saul@senecacollege.ca < message.txt** | Sends the contents of the file **message.txt** as an email to the recipient **murray.saul@senecacollege.ca** with the subject line "Important Message" |

Week 8

| **Command** | **Useful Options** | **Purpose** |
| --- | --- | --- |
| ln | -s, -f | Creates links between files (hard or symbolic) |
| ps | -e, -f, -u | Displays a list of currently running processes |
| top | -d, -u, -p | Provides a dynamic real-time view of system processes and their resource usage |
| fg | %job\_id, %process\_id | Brings a suspended or backgrounded process back to the foreground |
| bg | %job\_id, %process\_id | Puts a suspended or stopped process in the background, allowing it to continue running |
| jobs | -l, -n | Lists all current jobs, including their job ID, status, and command |
| kill | -9, -15, -SIGKILL, -SIGTERM | Sends a signal to terminate a process or job |
| alias | -p | Creates a shortcut (alias) for a command or a sequence of commands |
| unalias | -a | Removes an alias previously defined with the **alias** command |
| history | -c, -d, -a | Displays a list of previously executed commands, including their line numbers, and allows for various manipulations of that list |

The & symbol at the end of the command is used to run a process in the background.