**Vi Editor- Assignments**

1. Invoke the vi editor with a filename (which is to be created)

**$ vi cityfile**

2. Type a few lines(at least 10) in the editor. Let one or more lines contain the words given below.

**delhi**

**mumbai**

**chennai Mumbai**

**Bangalore**

**Gandhinagar**

**Gurgoan**

**Kolkata**

**Coimbatore**

**Hyderabad**

**Pune**

3.Save the file without exiting the editor

**Press ESC then type colon(:) . type w after colon and press enter this will save the file with exiting.**

4. Navigate quickly to the beginning & end of the file

**Esc then g - for the beginning of file Capital G for ending of file**

5. Delete first two lines & copy it after the last line.

**Press ESC move the cursor to the line we want to remove and press**

**[2]dd and 2yy**

6. Now Copy the last two lines & paste them before the first

**Place The cursor at last and press 2yy and press Capital P before the first line to paste the contents before the cursor**

7. Combine last two lines of the file in a single line.

**Position the cursor click Capital J to join the the cursor is on with the below one**

8. How will you insert a line before the first line?

**Add \n at the end of the string**

9. How will you paste a yanked buffer before the current line?

**Place the cursor at the current line and press p.**

10. Search the document for a string "delhi"

**/delhi**

11. Replace a single occurrence of "delhi" with "Mumbai"

**/delhi**

**cw Mubai**

12. Replace all occurrences of "delhi" on a particular line with "Mumbai" using one command

**s/delhi/Mumbai/g**

13. Replace "delhi" with "mumbai" for the entire document using one command.

**1,$s/delhi/mumbai/g**

14. Replace all occurrences of "delhi" with "mumbai" for part of a document (e.g.

line number 13 to 37).

**13,37/s/delhi/mumbai/g**

15. Undo the above command

**:u**

16. Replace only the first occurrence of "delhi" with "mumbai" in each line, for part

of a document (e.g. line number 13 to 37).

**13,37/s/delhi/mumbai**

17. Undo the above comment. Redo it. Undo it again.

**:u then ctrl-r then :u**

18. Replace all occurrences of "delhi" with "mumbai" for complete file interactively.

19. Search for a word backwards. Loop through all matches.

**Press ? then type the search word**

20. Search for a word forwards. Loop through all matches

**Press / then types the search word**

21. Join two lines together.

**Place the cursor at the required line and press J to join the 2 lines**

22. Enable line numbering and then disable it

**:set nu – enabling**

**:set nu! - disabling**

23. Make the search case insensitive

**:set ic for search case insensitive**

24. Delete 4 lines (and then paste them somewhere else)

**4yy then 4dd then P**

25. What is the command to delete part of the line starting from the current cursor

position?

**::d**

26. What is the command to delete the current word?

**Place the cursor front and press Ctrl+w**

**GCC**

1. Log into the Linux server using your respective ids

**Sudo -i**

**Password:**

1. Copy the C source file simple\_program.c. Compile the file with the following command:

**gcc simple.c**

Where *sourcefilename.c* is the name of the C source code

1. Observe the executable formed is called a.out. Execute the file using the command:

**./a.out**

1. Once again compile the source code with the following command:

**gcc –o exec.out simple.c**

Where *outputfilename* is the name of the executable file. After this command gets executed successfully, the name of the executable is not a.out but whatever is given as *outputfilename*

Execute the output file using the command:

./outputfilename

## Object files

1. Once again compile the source code with the following command:

**gcc –c simple.c**

Observe using ls that an object file called *sourcefilename.o* is created in the directory

What is the difference between an executable and an object file?

Now, create the executable from the object file using the following command:

gcc *sourcefilename.o*

## Additional gcc switches

1. Copy the program program\_warn.c as directed by the facilitator. Compile the program using gcc without any switch. Observe the result of compilation

**Compiled Succesfully**

1. Now, compile the program with the –Wall switch. Observe the warnings that are now given by the compiler.

**Warning messages gets enabled by gcc -Wall**

1. Now, compile the program with the –ansi switch. Observe the warnings that are now given by the compiler.

**gcc -Ansi will show primitive Ansi C multiple line comments related warnings**

## Linking

1. Write a program linking1.c, which has main function. Inside main there should be a call to another function which is defined in another file linking2.c

**vi linking1.c**

**#include”swapstr.h”**

**int main()**

**{**

**char n1[10];**

**char n2[10];**

**printf(“Enter n1:”);**

**scanf(“%s”.n1);**

**printf(“Enter n2:”);**

**scanf(“%s”.n2);**

**printf(“old string:”);**

**printf(“n1 : %n\n”,n1);**

**printf(“n2 : %n\n”,n2);**

**swapstr(s1,s2);**

**printf(“new string:”);**

**printf(“n1 : %n\n”,n1);**

**printf(“n2 : %n\n”,n2);**

**}**

**vi linking2.c**

**#include “swapstr.h”**

**Void swapstr(char x[10] , char y[10])**

**{**

**char temp[10];**

**strcpy(temp,x);**

**strcpy(x,y);**

**strcpy(y,temp);**

**}**

**vi.linking2.h**

**#include<stdio.h>**

**#include<string.h>**

**void swapstr( char x[10] , char y[10]);**

1. Compile the file linking1.c using the following command:

gcc –c –Wall linking1.c

**int y; (unused variable)**

Again compile the file linking1.c using the following command:

gcc –Wall linking1.c

Observe the errors received

1. Now create a header file prototype.h, which contains the prototype of the function defined in linking2.c. Let the file content begin and end with lines as below, to avoid multiple inclusion of .h file.

**#ifndef \_\_PROTOTYPE\_H\_**

**#define \_\_PROTOTYPE\_H\_**

**void func();** // an example declaration of function defined in prototype.h

**#endif** //end of \_\_PROTOTYPE\_H\_

1. Include this file in linking1.c using the following statement:

**gcc –I. –c –Wall linking1.c**

1. Similar effect can be achieved by including the file prototype.h using the following

command

**gcc –c –Wall linking1.c**

1. Now after all the compilation warnings have been removed, link the 2 files together:

**gcc –c –Wall linking1.c**

**gcc –c –Wall linking2.c**

**gcc linking1.o linking2.o**

1. Do not include the header file prototype.h containing the prototype of the function in the file linking1.c.

Instead of that include the file linking2.c in the file linking1.c using the following statement:

**gcc –c –Wall linking1.c**

**gcc –c –Wall linking2.c**

**gcc linking1.o linking2.o**

1. Do not include the file linking2.c in linking1.c. Instead of that include the file prototype.h and proceed as mentioned in the points 15 and 16.

**#include<prototype.h>**