Aim:

Execution of Unix General Purpose UtilityCommands

Theory:

The UNIX operating system has for many years formed the backbone of the Internet, especially for large servers and most major university campuses. However, a free version of UNIX called Linux has been making significant gains against Macintosh and the Microsoft Windows 95/98/NT environments, so often associated with personal computers. Developed by a number of volunteers on the Internet such as the Linux group and the GNU project, much of the open-source software is copyrighted, but available for free. This is especially valuable for those in educational environments where budgets are often limited.

UNIX commands can often be grouped together to make even more powerful commands with capabilities known as I/O redirection ( < for getting input from a file input and > for outputing to a file ) and piping using | to feed the output of one command as input to the next. Please investigate manuals in the lab for more examples than the few offered here.

TTY

tty - print the file name of the terminal connected to standard input

Print the file name of the terminal connected to standard input.

-s, --silent, --quiet print nothing, only return an exit status

--help display this help and exit

--version output version information and exit

Output

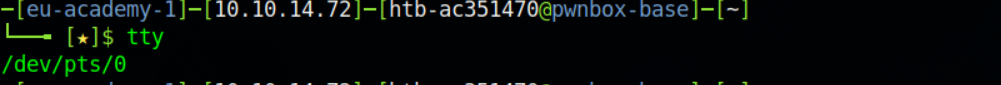


Fig 2.1

Echo

DESCRIPTION

Echo the STRING(s) to standard output

-n do not output the trailing newline

-e enable interpretation of backslash escapes

-E disable interpretation of backslash escapes (default)

--help display this help and exit

--version output version information

Output

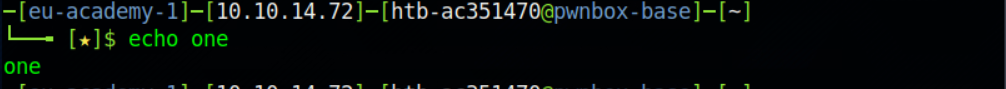


Fig 2.2

Ping

DESCRIPTION

ping uses the ICMP protocol's mandatory ECHO\_REQUEST datagram to elicitan ICMP ECHO\_RESPONSE from a host or gateway. ECHO\_REQUEST datagrams (“pings”) have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of “pad” bytes used to fill out the packet.ping works with both IPv4 and IPv6. Using only one of them explicitly can be enforced by specifying -4 or -6. ping can also send IPv6 Node Information Queries (RFC4620). Intermediate hops may not be allowed, because IPv6 source routing was deprecated (RFC5095).

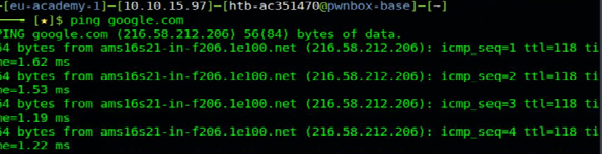


Fig 2.3

Clear

Clears your screen if this is possible, including its scrollback

buffer (if the extended “E3” capability is defined). clear looks in

the environment for the terminal type given by the environment variable

TERM, and then in the terminfo database to determine how to clear the

screen. clear writes to the standard output. You can redirect the standard

output to a file (which prevents clear from actually clearing the

screen), and later cat the file to the screen, clearing it at that

point.

Exit

The exit() function causes normal process termination and the least

significant byte of status (i.e., status & 0xFF) is returned to the parent (see wait(2)).All functions registered with atexit(3) and on\_exit(3) are called, in

the reverse order of their registration. (It is possible for one of these functions to use atexit(3) or on\_exit(3) to register an additional function to be executed during exit processing; the new regitration is added to the front of the list of functions that remain tobe called.) If one of these functions does not return (e.g., it calls exit(2), or kills itself with a signal), then none of the remaining functions is called, and further exit processing (in particular, flush‐ing of stdio(3) streams) is abandoned. If a function has been registered multiple times using atexit(3) or on\_exit(3), then it is called as many times as it was registered.

Ifconfig

Ifconfig is used to configure the kernel-resident network interfaces.

It is used at boot time to set up interfaces as necessary. After that,

it is usually only needed when debugging or when system tuning is

needed. If no arguments are given, ifconfig displays the status of the cur‐

rently active interfaces. If a single interface argument is given, it

displays the status of the given interface only; if a single -a argu‐

ment is given, it displays the status of all interfaces, even those

that are down. Otherwise, it configures an interface.

Output

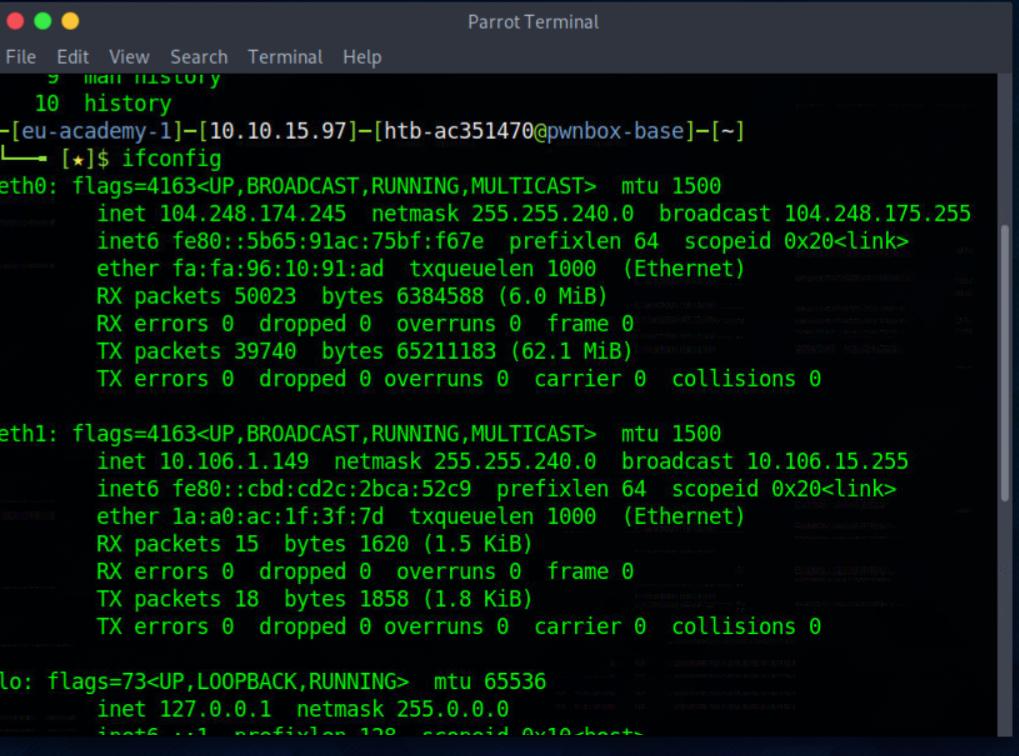


Fig 2.4

PWD

Print the full filename of the current working directory

-L, --logical use PWD from environment, even if it contains symlinks

-P, --physical avoid all symlinks

--help display this help and exit

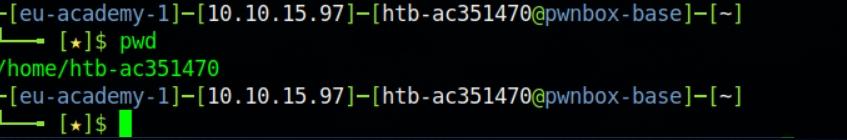
Output

Fig 2.5

Time

Display the current time in the given FORMAT, or set the system date. Mandatory arguments to long options are mandatory for short options too.

-d, --date=STRIN display time described by STRING, not 'now'

--debug annotate the parsed date, and warn about questionable usage to stderr

-f, --file=DATEFILE like --date; once for each line of DATEFILE

-I[FMT], --iso-8601[=FMT]

output date/time in ISO 8601 format. FMT='date' for date only (the default), 'hours', 'minutes', 'seconds', or 'ns' for date and time to the indicated precision. Example: 2006-08-14T02:34:56-06:00 -R, --rfc-email

--rfc-3339=FMT

output date/time in RFC 3339 format. FMT='date', 'seconds', or 'ns' for date and time to the indicated precision. Example: 2006-08-14 02:34:56-06:00

-r, --reference=FILE display the last modification time of FILE

-s, --set=STRING set time described by STRING

-u, --utc, --universal print or set Coordinated Universal Time (UTC)

Uptime

uptime gives a one line display of the following information. The current time, how long the system has been running, how many users are currently logged on, and the system load averages for the past 1, 5, and 15 minutes .This is the same information contained in the header line displayed by w(1)System load averages is the average number of processes that are either in a runnable or uninterruptable state. A process in a runnable state is either using the CPU or waiting to use the CPU. A process in uninterruptable state is waiting for some I/O access, eg waiting for disk. The averages are taken over the three time intervals. Load averages are not normalized for the number of CPUs in a system, so a load average of 1 means a single CPU system is loaded all the time while on a 4 CPU system it means it was idle 75% of the time.

Output

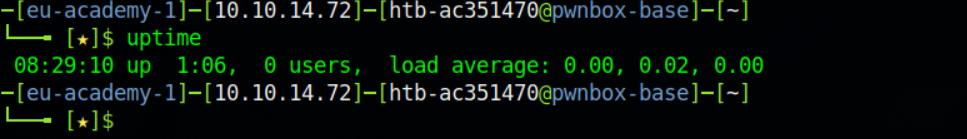


Fig 2.6

Man

man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections. The table below shows the section numbers of the manual followed by the types of pages they contain.

1 Executable programs or shell commands

2 System calls (functions provided by the kernel)

3 Library calls (functions within program libraries)

4 Special files (usually found in /dev)

5 File formats and conventions, e.g. /etc/passwd

6 Games

7 Miscellaneous (including macro packages and conventions)

8 System administration commands (usually only for root)

9 Kernel routines [Non standard]

Output

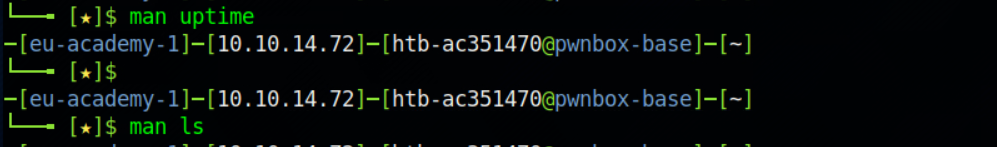


Fig 2.7

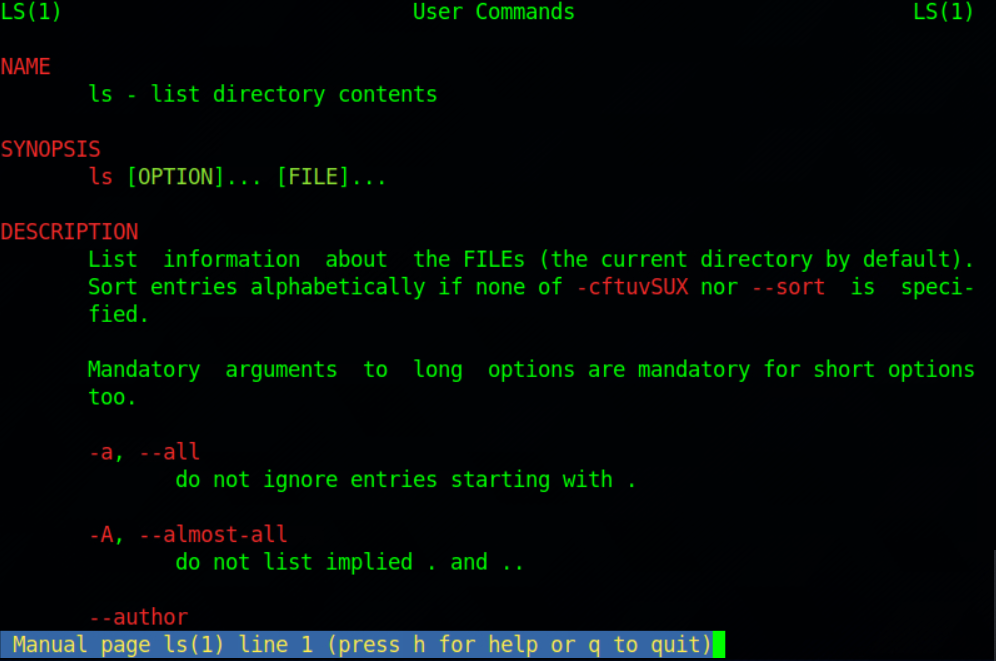


Fig 2.8

Cat

Concatenate FILE(s) to standard output. With no FILE, or when FILE is -, read standard input.

-A, --show-all

-b, --number-nonblank

-e equivalent to –vE

-E, --show-ends display $ at end of each line

-n, --number number all output lines

-s, --squeeze-blank suppress repeated empty output lines

-t equivalent to -vT

-T, --show-tabs display TAB characters as ^I

-u (ignored)

-v, --show-nonprinting use ^ and M- notation, except for LFD and TAB

Output

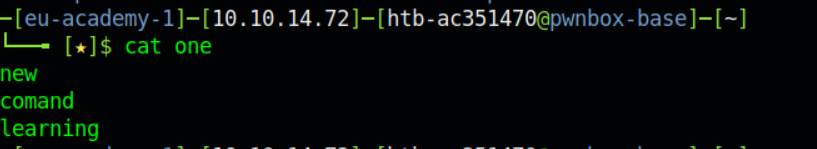


Fig 2.9

Which

which returns the pathnames of the files (or links) which would be executed in the current environment, had its arguments been given as commands in a strictly POSIX-conformant shell. It does this by searching the PATH for executable files matching the names of the arguments. It does not canonicalize path names.

Output

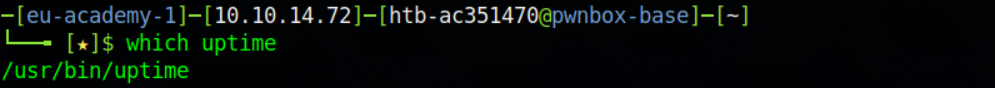


Fig 2.10

ID

Print the user name associated with the current effective user ID. Same as id -un.

--help display this help and exit

--version output version information and exit Paginate or columnate FILE(s) for printing.

With no FILE, or when FILE is -, read standard input. Mandatory arguments to long options are mandatory for short options too.

+FIRST\_PAGE[:LAST\_PAGE], --pages=FIRST\_PAGE[:LAST\_PAGE]begin [stop] printing with page FIRST\_[LAST\_]PAGE

-a, --across print columns across rather than down, used together with -

-c, --show-control-chars use hat notation (^G) and octal backslash notation

-d, --double-space double space the output

-D, --date-format=FORMAT use FORMAT for the header date

-e[CHAR[WIDTH]], --expand-tabs[=CHAR[WIDTH]] expand input CHARs (TABs) to tab WIDTH (8)

-F, -f, --form-feed use form feeds instead of newlines to separate pages (by a 3-line page header with -F or a 5-line header and trailer without -F)

-h, --header=HEADER use a centered HEADER instead of filename in page header, -h "" prints a blank line, don't use -h""

-J, --join-lines merge full lines, turns off -W line truncation, no column alignment, --sep-string[=STRING] sets separators

Output

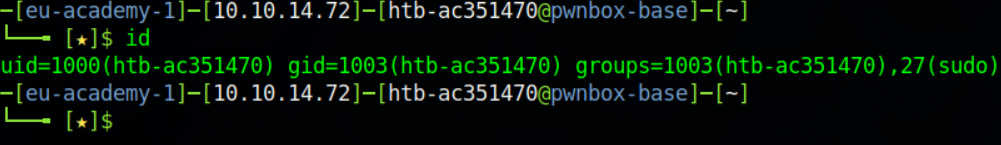


Fig 2.11

Conculsion:

In the above experiment we studied about various linux commands and also performed them and there are many more such commands more left to explore .

2.B

Aim: To study VI and Gedit

Theory

Vi editor

The vi editor is elaborated as visual editor. It is installed in every Unix system. In other words, it is available in all Linux distros. It is user-friendly and works same on different distros and platforms. It is a very powerful application. An improved version of vi editor is vim.

The vi editor has two modes:

Command Mode: In command mode, actions are taken on the file. The vi editor starts in command mode. Here, the typed words will act as commands in vi editor. To pass a command, you need to be in command mode.

Insert Mode: In insert mode, entered text will be inserted into the file. The Esc key will take you to the command mode from insert mode.

By default, the vi editor starts in command mode. To enter text, you have to be in insert mode, just type 'i' and you'll be in insert mode. Although, after typing i nothing will appear on the screen but you'll be in insert mode. Now you can type anything.

To exit from insert mode press Esc key, you'll be directed to command mode.

If you are not sure which mode you are in, press Esc key twice and you'll be in command mode.

Advantages

1. Vim is Free and Open Source

2. Vim Is Well Documented

3. Vim Is Very Customizable and Extensible

Disadvantages

1.Very complex to use not beginner friendly

2.Need to remember some commands to do task

Output : This the output using vi editor

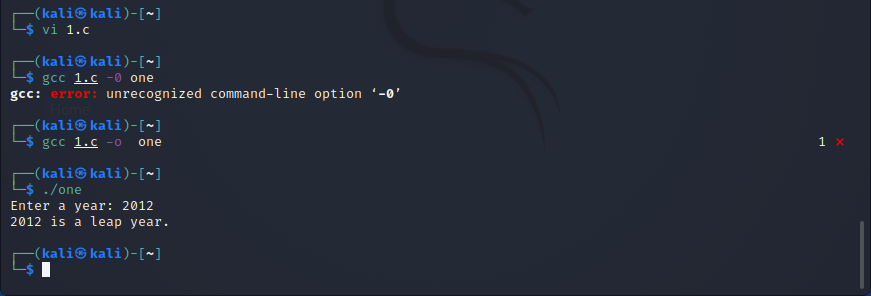


Fig 2.12

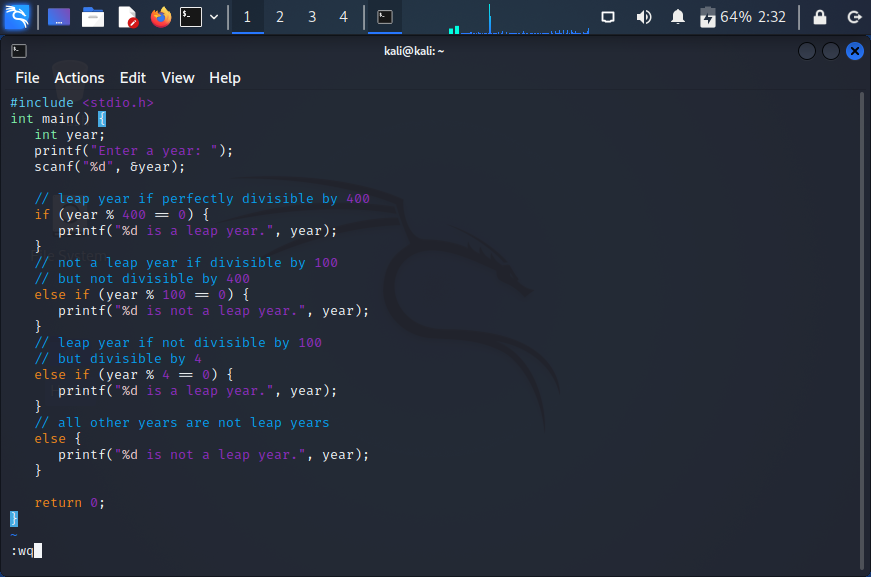


Fig 2.13

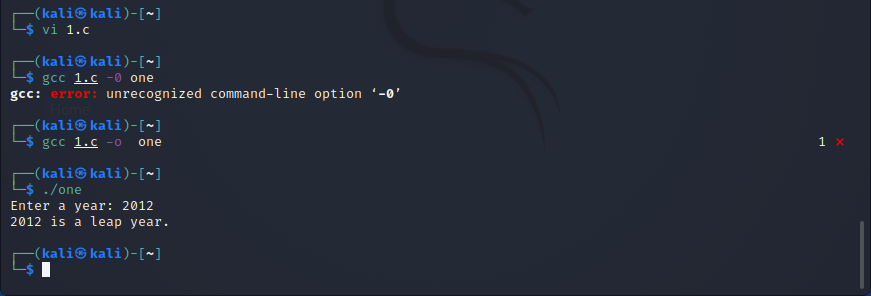


Fig 2.14

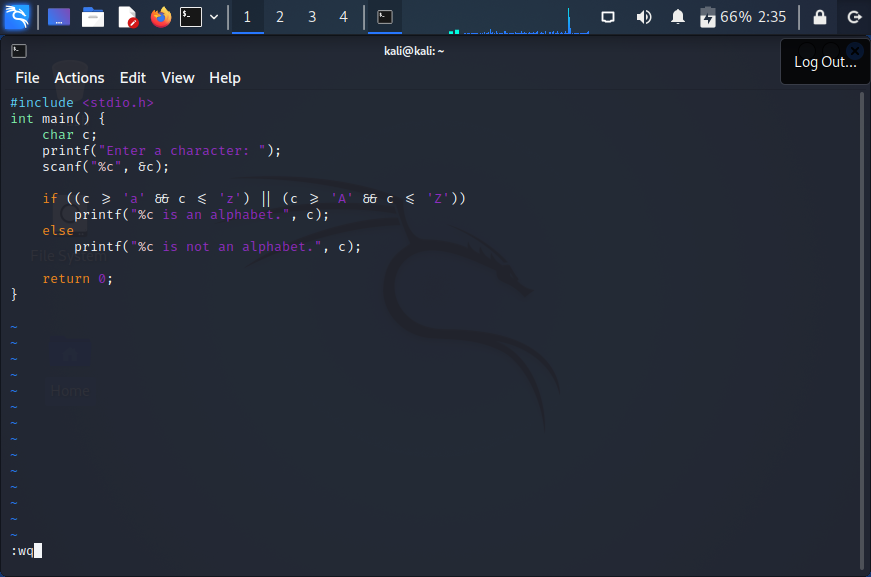


Fig 2.15

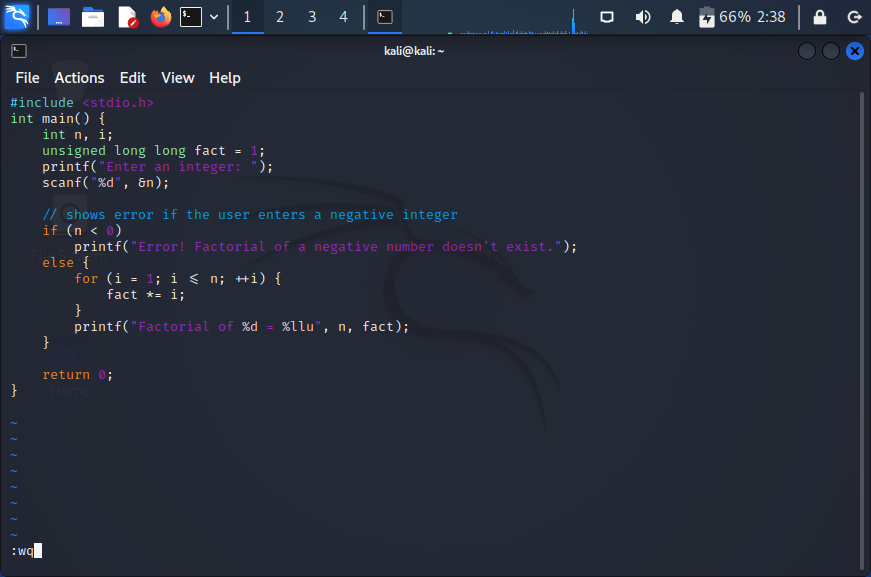
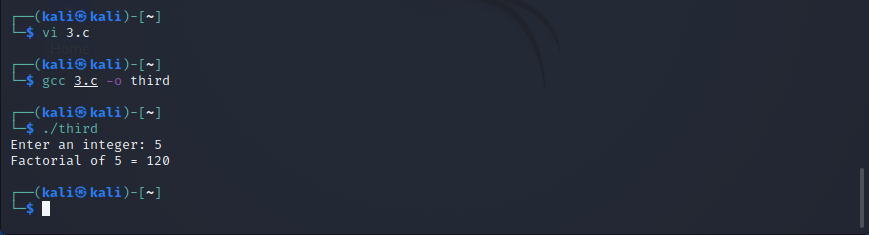


Fig 2.16



Gedit

Gedit is a Linux text editor commonly deployed as part of the GNOME desktop environment. While most Linux guides and tutorials recommend using the nano editor or vi to edit text files and configuration files, the gEdit editor is easier to use than nano and vi and works the same as Microsoft Windows Notepad.

The gEdit user interface contains a single toolbar at the top with a panel for entering the text below it.

The toolbar contains items to open and save files, add tabs, and control the window. Select Open to display a window with a search bar to search for documents, a list of recently accessed documents, and the option to access Other Documents. When you choose Other Documents, a file dialog appears where you can search the directory structure for the file you want to open.

There is a plus symbol (+) next to Open. When you select it, a new tab is added so that you can edit multiple documents at the same time.

Save displays the File dialog. Choose a location in the file system to save the file. You can also choose the character encoding and the file type.The Options icon is denoted by three vertical lines. When selected, it opens a new menu with options to refresh the screen, save and print a document, find information in a document, change the view, and more.

Advantages

1.Very easy to use beginner friendly

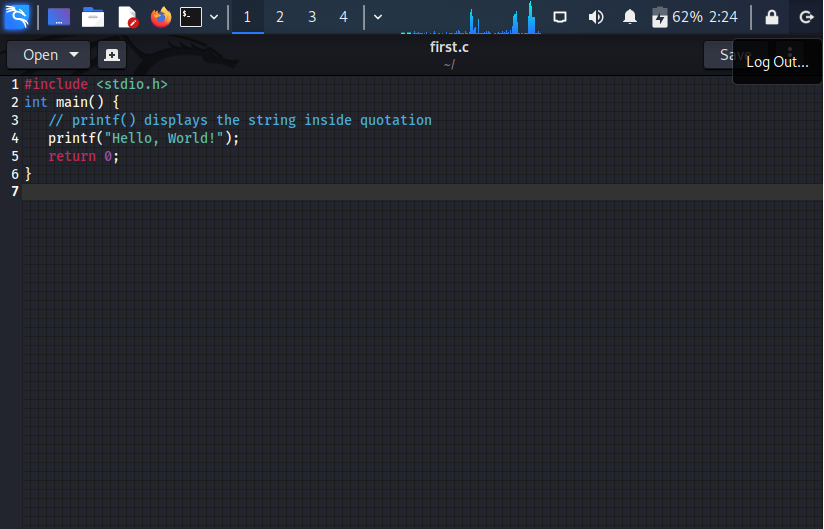
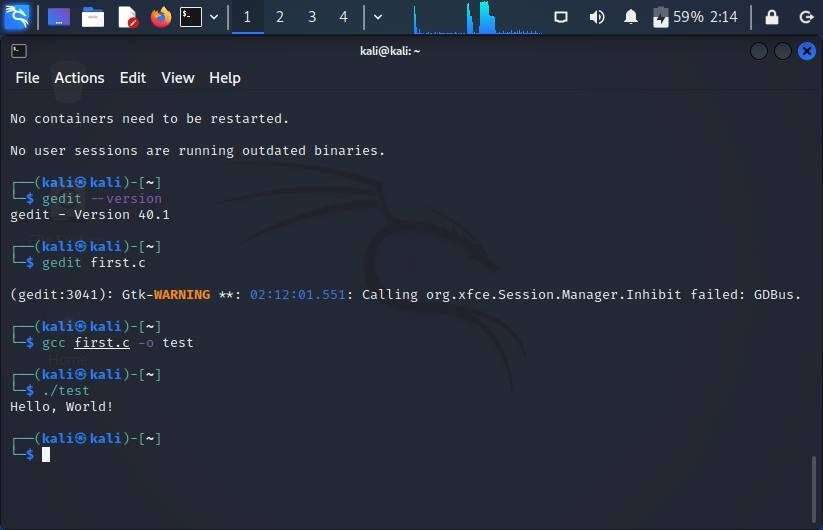
2.Compatable in many languages

3.Clipboard support

Disadvantages

The user is always kept guessing. There are no self-explanatory error messages. If anything goes wrong no error messages appear, only the speaker beeps to inform you that something went wrong. There are three modes in which the editor works. Under each mode the same keypress creates different effects.

Output :this is the output with gedit



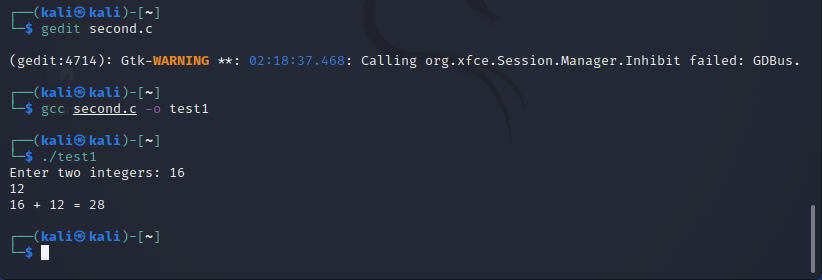
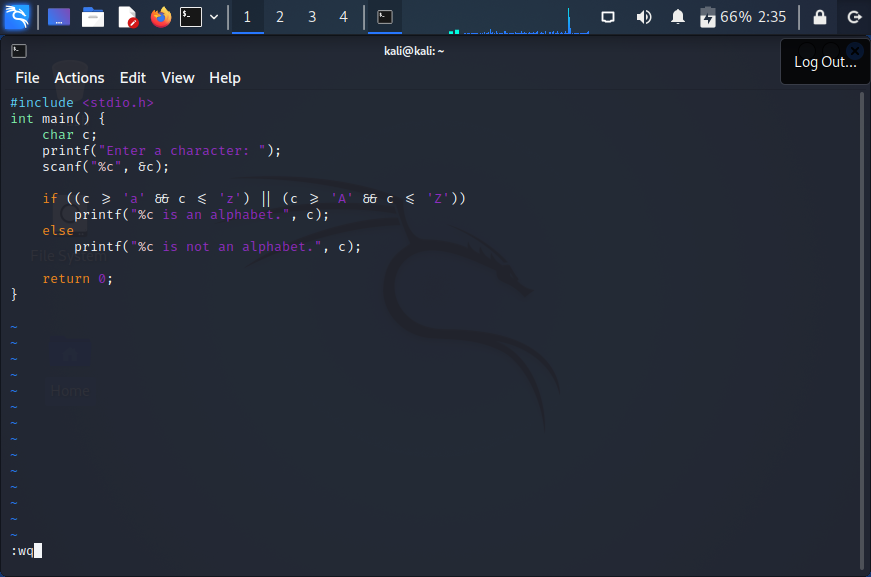


Fig 2.16



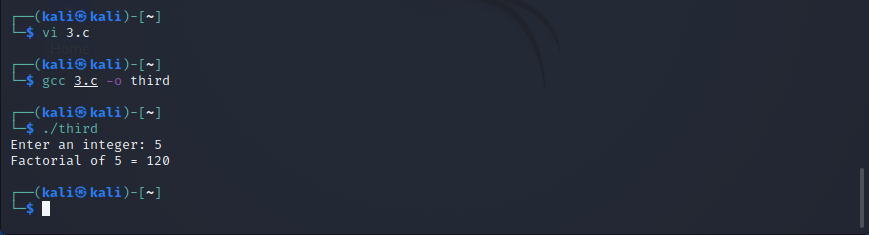


Fig.2.17

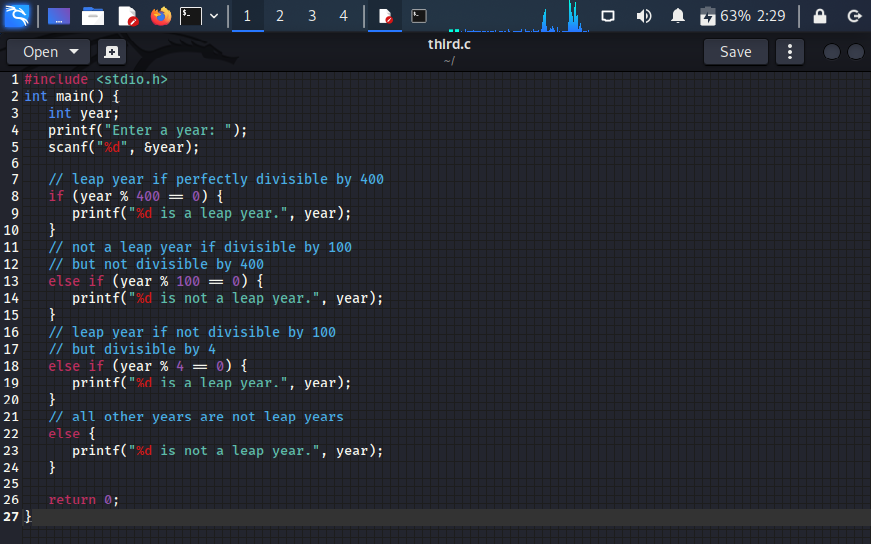


Fig 2.18

Conculsion :

In the above experiment we learned about how to use vi editter and gedit and all the different modes in this editer vi is open soure free in build but nanot so beginer friendly on the side in some machines we have to install gedit but is is very easy to use one should learmn both editors.