

Ex 3

COMPILING FROM THE SOURCE

Date: 29.08.20

Aim:

To study and implement the compiling from the source.

Description:

tar

The Linux ‘tar’ stands for tape archive, is used to create Archive and extract the Archive files. tar command in Linux is one of the important commands which provides archiving functionality in Linux. We can use Linux tar command to create compressed or uncompressed Archive files and also maintain and modify them.

zip

ZIP is a compression and file packaging utility for Unix. Each file is stored in a single .zip { .zip-filename } file with the extension .zip. ZIP is a compression and file packaging utility for Unix. Each file is stored in a single .zip { .zip-filename } file with the extension .zip.

gzip

gzip command compresses files. Each single file is compressed into a single file. The compressed file consists of a GNU zip header and deflated data. If given a file as an argument, gzip compresses the file, adds a “.gz” suffix, and deletes the original file. With no arguments, gzip compresses the standard input and writes the compressed file to standard output.

Difference between Gzip and zip command in Unix and when to use which command

- ZIP and GZIP are two very popular methods of compressing files, in order to save space, or to reduce the amount of time needed to transmit the files across the network, or internet.
- In general, GZIP is much better compared to ZIP, in terms of compression, especially when compressing a huge number of files.
- The common practice with GZIP, is to archive all the files into a single tarball before compression. In ZIP files, the individual files are compressed and then added to the archive.

18CS2064 - Open Source technologies Lab | URK17CS293

- When you want to pull a single file from a ZIP, it is simply extracted, then decompressed. With GZIP, the whole file needs to be decompressed before you can extract the file you want from the archive.
- When pulling a 1MB file from a 10GB archive, it is quite clear that it would take a lot longer in GZIP, than in ZIP.
- GZIP's disadvantage in how it operates, is also responsible for GZIP's advantage. Since the compression algorithm in GZIP compresses one large file instead of multiple smaller ones, it can take advantage of the redundancy in the files to reduce the file size even further.
- If you archive and compress 10 identical files with ZIP and GZIP, the ZIP file would be over 10 times bigger than the resulting GZIP file.

Commands:

Sl. No.	Command Name	Syntax	options
1.	rpm	rpm {rpm-file}	-a, --all Query all packages -f Query for packages owning given file
2.	apt-get	apt-get [options] source pkg1 [pkg2 ...]	-a It prints all the system information in the order -s It prints the kernel name. -n It prints the hostname of the network node -r It prints the kernel release

			<p>date</p> <p>-v</p> <p>It prints the version of the current kernel</p>
3.	tar	tar [options] [archive-file] [file or directory to be archived]	<p>-c</p> <p>Creates Archive</p> <p>-x</p> <p>Extract the archive</p> <p>-f</p> <p>Creates archive with given filename</p> <p>-t</p> <p>Displays or lists files in archive file</p> <p>-u</p> <p>Archives and adds to an existing archive file</p> <p>-A</p> <p>Concatenates the archive files</p> <p>-z</p> <p>zip, tells tar command that create tar file using gzip</p> <p>-W</p> <p>Verify a archive file</p> <p>-r</p> <p>update or add file or directory in already existed .tar file</p>

Exercise:

1. Compile the source from Openttd package

Installing Openttd using package manager

```
~/Downloads sudo apt-get install openttd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fluid-soundfont-gm libao-common libao4 libxdg-basedir1 openttd-data
  openttd-opengfx openttd-openmsx timidity
Suggested packages:
  fluid-soundfont-gs fluidsynth libsndio6.1 openttd-opensfx freepats pmidi
  timidity-daemon
The following NEW packages will be installed:
  fluid-soundfont-gm libao-common libao4 libxdg-basedir1 openttd openttd-data
  openttd-opengfx openttd-openmsx timidity
0 upgraded, 9 newly installed, 0 to remove and 544 not upgraded.
Need to get 128 MB of archives.
After this operation, 175 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 fluid-soundfont-gm all 3.1-5.1 [120 MB]
Get:2 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 libao-common all 1.2.2+20180113-1 [11.6 kB]
Get:3 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 libao4 amd64 1.2.2+20180113-1+b1 [37.6 kB]
```

Check the version of Openttd

```
~/Downloads openttd version
OpentTD 1.10.3
```

Output:

Run the Openttd



2. Compile the source from JFrog package

Download and Extract the JFrog file

```
~ cd Downloads ✓
~/Downloads mkdir jfrog_lab ✓
~/Downloads mv jfrog-artifactory-oss-7.7.3-linux.tar.gz jfrog_lab ✓
~/Downloads cd jfrog_lab ✓
~/Downloads/jfrog_lab tar -xvf jfrog-artifactory-oss-7.7.3-linux.tar.gz
artifactory-oss-7.7.3/app/metadata/bin/jf-metadata
artifactory-oss-7.7.3/app/router/bin/jf-router
artifactory-oss-7.7.3/app/doc/Frontend-About-Box.html
artifactory-oss-7.7.3/var/
artifactory-oss-7.7.3/var/bootstrap/
artifactory-oss-7.7.3/var/bootstrap/access/
artifactory-oss-7.7.3/var/bootstrap/access/etc/
artifactory-oss-7.7.3/var/bootstrap/access/etc/security/
```


Compiling the Source of JFrog

```
./.artifactoryctl
2020-09-13T07:55:34.682Z [jfrt ] [INFO ] [fd2d4d5c0f3a2dc4] [SchemaInitializatio
nManager:51] [localhost-startStop-1] - Post-DB initialization manager initialized
2020-09-13T07:55:36.988Z [jfrt ] [INFO ] [86c72b1a103ddcef] [ctoryContextConfigL
istener:325] [art-init ] -

Version: 7.7.3

~/.Downloads/jfrog_lab cd artifactory-oss-7.7.3
~/Dow/j/artifactory-oss-7.7.3 cd app
~/Dow/j/a/app cd bin
~/Dow/j/a/app/bin ./artifactoryctl
2020-09-13T07:54:35.966Z [shell] [INFO ] [] [installerCommon.sh:1334 ] [ma
in] - Checking open files and processes limits
2020-09-13T07:54:35.989Z [shell] [INFO ] [] [installerCommon.sh:1337 ] [ma
in] - Current max open files is 1024
2020-09-13T07:54:36.017Z [shell] [INFO ] [] [installerCommon.sh:1348 ] [ma
in] - Current max open processes is 11582
2020-09-13T07:54:36.053Z [shell] [INFO ] [] [installerCommon.sh:1415 ] [ma
in] - Testing directory /home/reeves/Downloads/jfrog_lab/artifactory-oss-7.7.3/v
ar has read/write permissions for user id 1001
2020-09-13T07:54:36.088Z [shell] [INFO ] [] [installerCommon.sh:1430 ] [ma
```

Run the JFrog in the Browser and Enter Valid Credentials



WELCOME TO JFROG

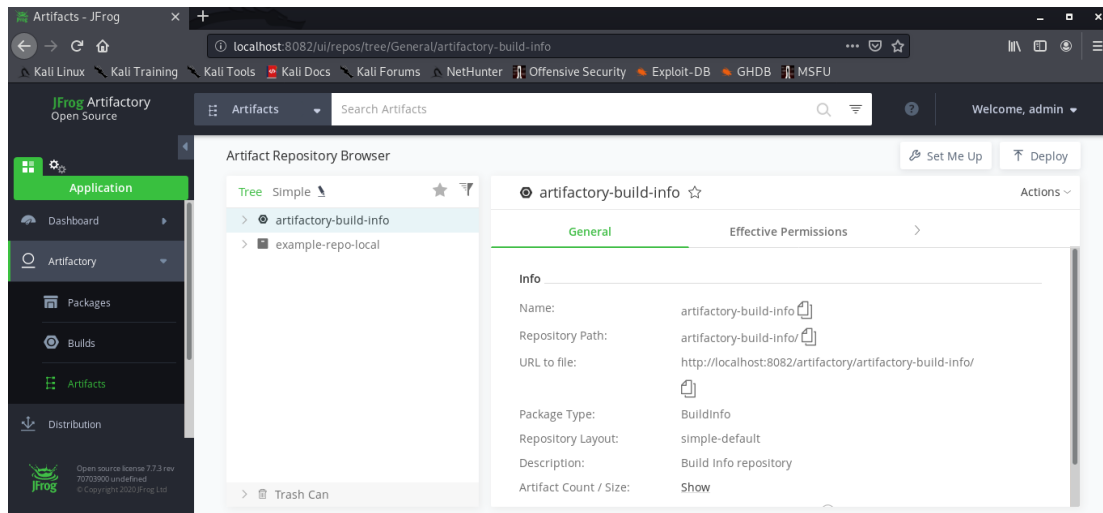
Username*

Password*

☐ Remember me

Login

Output:



3. Compile the Source from gcc

Check the Version of gcc

```
gcc --version
gcc (Debian 9.3.0-15) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Sample Program

```
vi hello.c

#include <stdio.h>
int main() {
    printf("Welcome to Open Source Technologies Lab \n");
    printf("This is Experiment-3");
}
```

Output:

Compile and Run the C program using gcc

```
~/Downloads vi hello.c
~/Downloads gcc -o intro hello.c
~/Downloads ./intro
Welcome to Open Source Technologies Lab
This is Experiment-3%
~/Downloads
```

4. Compile the source from any open source package(GIMP)

Installing GIMP using package manager

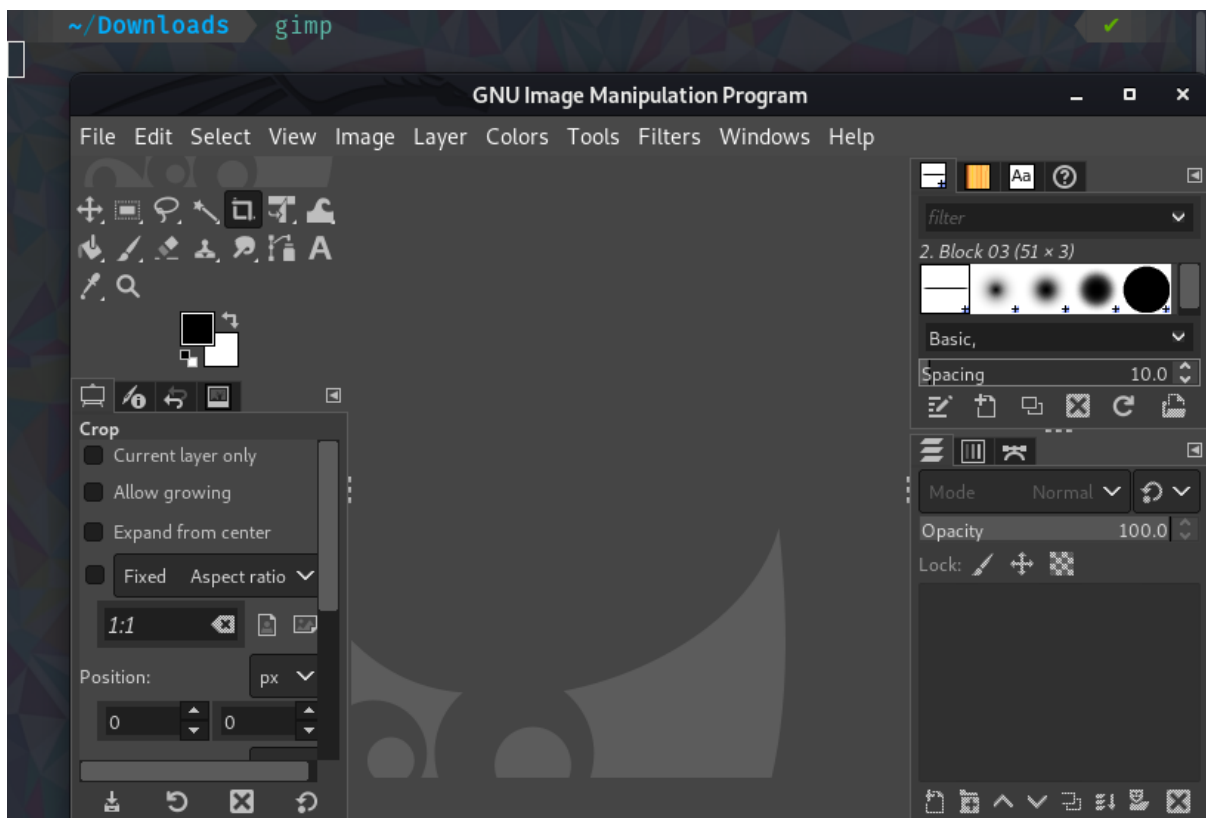
```
~/Downloads sudo apt-get install gimp -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ghostscript gimp-data gsfonds libamd2 libbabl-0.1-0 libcamd2 libccolamd2
  libcholmod3 libcolamd2 libgegl-0.4-0 libgegl-common libgimp2.0 libgs9
  libgs9-common libheif1 libilmbase25 libjson-c5 libmetis5 libmng1
  libmypaint-1.5-1 libmypaint-common libopenexr25 libraw20
  libsuitesparseconfig5 libumfpack5 libwmf0.2-7
Suggested packages:
  ghostscript-x gimp-help-en | gimp-help gimp-data-extras libwmf0.2-7-gtk
The following NEW packages will be installed:
  ghostscript gimp gimp-data gsfonds libamd2 libbabl-0.1-0 libcamd2
  libccolamd2 libcholmod3 libcolamd2 libgegl-0.4-0 libgegl-common libgimp2.0
  libheif1 libilmbase25 libjson-c5 libmetis5 libmng1 libmypaint-1.5-1
  libmypaint-common libopenexr25 libraw20 libsuitesparseconfig5 libumfpack5
  libwmf0.2-7
The following packages will be upgraded:
  libgs9 libgs9-common
2 upgraded, 25 newly installed, 0 to remove and 540 not upgraded.
Need to get 38.3 MB of archives.
After this operation, 145 MB of additional disk space will be used.
```


Update the System to Compile GIMP

```
~/Downloads sudo apt-get update
Get:1 http://ftp.halifax.rwth-aachen.de/kali kali-rolling InRelease [30.5 kB]
Get:2 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main Sources [13.1 MB]
Get:3 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/contrib Sources [61.6 kB]
Get:4 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/main amd64 Packages [16.6 MB]
Get:5 http://ftp.halifax.rwth-aachen.de/kali kali-rolling/contrib amd64 Packages [99.8 kB]
Fetched 29.9 MB in 6min 22s (78.1 kB/s)
Reading package lists... Done
```

Output:

Run the GIMP



Results:

The compiling from the source is studied and executed.

Video Link:

<https://youtu.be/iYaqsdF99w4>