18CS2064 - Open Source technologies Lab | URK17CS023

Ex 3 COMPILING FROM THE SOURCE Date: 29.08.20

Aim:

To study and implement the compiling from the source.

Description:

tar Command :

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.

Syntax :

tar [options] [archive-file] [file or directory to be archived]

Options:

-c : Creates Archive

-x : Extract the archive

-f : creates archive with given filename

-t : displays or lists files in archive file

-u : archives and adds to an existing archive file

-v : Displays Verbose Information

-A : Concatenates the archive files

-z : zip, tells tar command that create tar file using gzip

-j : filter archive tar file using tbzip

-W : Verify a archive file

-r : update or add file or directory in already existed .tar file

zip Command :

18CS2064 - Open Source technologies Lab | URK17CS023

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 used to compress the files to reduce file size and also used as file package utility. zip is available in many operating systems like unix, linux, windows etc. ● If you have a limited bandwidth between two servers and want to transfer the files faster, then zip the files and transfer.
- The zip program puts one or more compressed files into a single zip archive, along with information about the files (name, path, date, time of last modification, protection, and check information to verify file integrity). An entire directory structure can be packed into a zip archive with a single command.
- Compression ratios of 2:1 to 3:1 are common for text files. zip has one compression method (deflation) and can also store files without compression. zip automatically chooses the better of the two for each file to be compressed. The program is useful for packaging a set of files for distribution; for archiving files; and for saving disk space by temporarily compressing unused files or directories.

Syntax :

zip [options] zipfile files_list

Syntax for Creating a zip file:

$zip myfile.zip filename.txt

Options :

-d : Removes the file from the zip archive

-u : Updates the file in the zip archive

-m : Deletes the original files after zipping

-r : To zip a directory recursively

-x : Exclude the files in creating the zip

-v : Verbose mode or print diagnostic version info

gzip Command :

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.
- 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.

Syntax :

gzip [Options] [filenames]

Options:

-f : Sometimes a file cannot be compressed

-k : By default when you compress a file using the “gzip” command you end up with a new file with the extension

-L : This option displays the gzip license

-r : This option can compress every file in a folder and its subfolders **-[1-9]** : It allows

to change the compression level

-v : his option displays the name and percentage reduction for each file compressed or decompressed

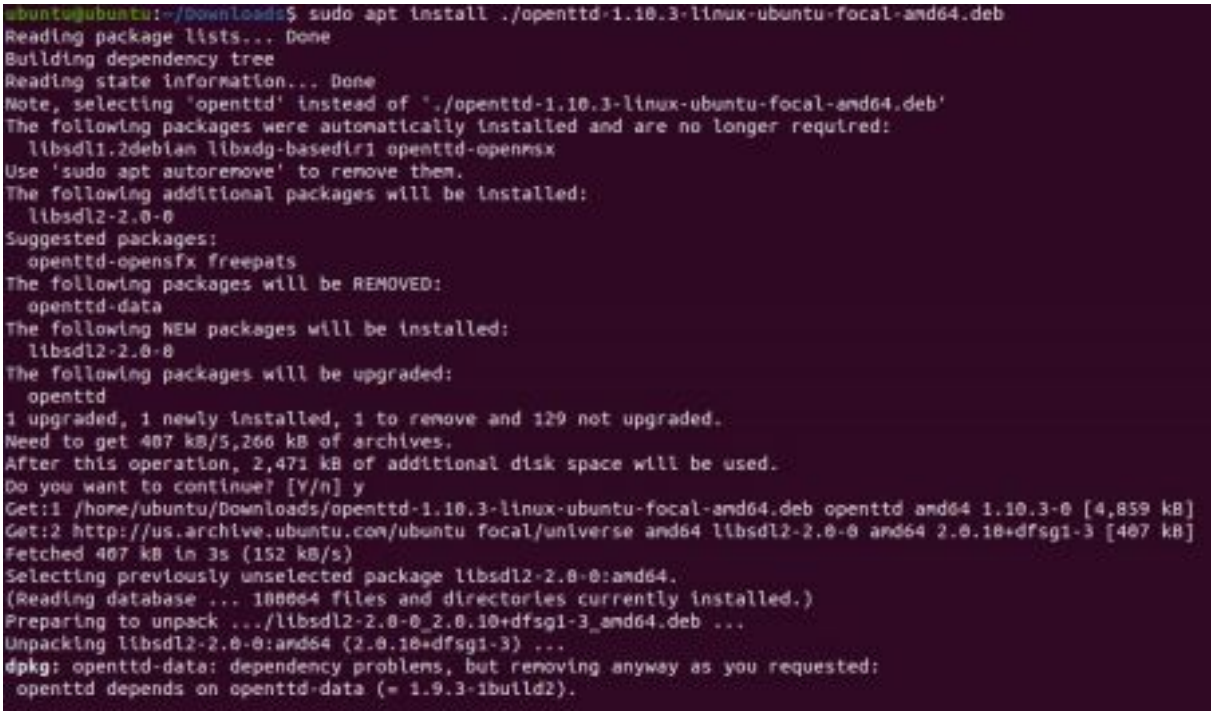
-d : This option allows you to decompress a file using the “gzip” command. **4**

18CS2064 - Open Source technologies Lab | URK17CS023

Exercise

1. Compile the source from openttd package

Installing openttd using dep(Debian) package



Compiling the source of openttd

2. Compile the source from JFrog package

Installing conan c++ package using dep(Debian) package

```
ubuntu@ubuntu:~/Downloads$ sudo apt install ./conan-ubuntu-64_1_29_0.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'conan' instead of './conan-ubuntu-64_1_29_0.deb'
The following packages were automatically installed and are no longer required:
  libsdl1.2debian libxdg-basedir1 openttd-openmsx
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential dpkg-dev fakeroot g++ g++-9
  gcc gcc-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan5
  libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0 libfakeroot
  libgcc-9-dev libitm1 liblsan0 libquadmath0 libstdc++-9-dev libtsan0 libubsan1 linux-libc-dev
  make manpages-dev
Suggested packages:
  binutils-doc debian-keyring g++-multilib g++-9-multilib gcc-9-doc gcc-multilib autoconf
  automake libtool flex bison gcc-doc gcc-9-multilib gcc-9-locales glibc-doc libstdc++-9-doc
  make-doc
The following NEW packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential conan dpkg-dev fakeroot g++
  g++-9 gcc gcc-9 libalgorithm-diff-perl libalgorithm-diff-xs-perl libalgorithm-merge-perl
  libasan5 libatomic1 libbinutils libc-dev-bin libc6-dev libcrypt-dev libctf-nobfd0 libctf0
  libfakeroot libgcc-9-dev libitm1 liblsan0 libquadmath0 libstdc++-9-dev libtsan0 libubsan1
  linux-libc-dev make manpages-dev
0 upgraded, 33 newly installed, 0 to remove and 129 not upgraded.
Need to get 31.4 MB/47.4 MB of archives.
After this operation, 166 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```


creating a sample package

```
ubuntu@ubuntu:~/Desktop/open source$ conan new Ex3/0.1
File saved: conanfile.py
ubuntu@ubuntu:~/Desktop/open source$ mkdir src
ubuntu@ubuntu:~/Desktop/open source$ cd src
ubuntu@ubuntu:~/Desktop/open source/src$ touch MyLib.h
ubuntu@ubuntu:~/Desktop/open source/src$ touch MyLib.cpp
ubuntu@ubuntu:~/Desktop/open source/src$ ls
MyLib.cpp  MyLib.h
ubuntu@ubuntu:~/Desktop/open source/src$ gedit MyLib.h
ubuntu@ubuntu:~/Desktop/open source/src$ gedit MyLib.cpp
ubuntu@ubuntu:~/Desktop/open source/src$ gedit MyLib.h
ubuntu@ubuntu:~/Desktop/open source/src$ touch CMakeLists.txt
ubuntu@ubuntu:~/Desktop/open source/src$ gedit CMakeLists.txt
ubuntu@ubuntu:~/Desktop/open source/src$ mkdir build
ubuntu@ubuntu:~/Desktop/open source/src$ cd build
```

Making up the package

```
ubuntu@ubuntu:~/Desktop/open source/src$ cmake ../src/
-- The C compiler identification is GNU 9.3.0
-- The CXX compiler identification is GNU 9.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/Desktop/open source/src
```


Build package output

```
ubuntu@ubuntu:~/Desktop/open source/src$ conan create . myself/MyLib
Exporting package recipe
MyLib/0.1@myself/MyLib: A new conanfile.py version was exported
MyLib/0.1@myself/MyLib: Folder: /home/ubuntu/.conan/data/MyLib/0.1/myself/MyLib/export
MyLib/0.1@myself/MyLib: Package recipe modified in export, forcing source folder removal
MyLib/0.1@myself/MyLib: Use the --keep-source, -k option to skip it
MyLib/0.1@myself/MyLib: Exported revision: e8aa540f83f3827d3af317087a2ac665
Configuration:
[settings]
arch=x86_64
arch_build=x86_64
build_type=Release
compiler=gcc
compiler.libcxx=libstdc++
compiler.version=9
os=Linux
os_build=Linux
[options]
[build_requires]
[env]

MyLib/0.1@myself/MyLib: Forced build from source
Installing package: MyLib/0.1@myself/MyLib
Requirements
    MyLib/0.1@myself/MyLib from local cache - Cache
Packages
    MyLib/0.1@myself/MyLib:2a30b7d6ea2202e5393ebda51c8729dd2162b9f8 - Build
```

3. Compile the source from gcc

Checking the version of gcc

```
ubuntu@ubuntu:~$ gcc --version
gcc (Ubuntu 9.3.0-10ubuntu2) 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.
```

Program :

```
#include <stdio.h>
int main() {
  printf("Welcome to Open Source Lab \n");
  return 0;
}
```

Output :

```
ubuntu@ubuntu:~$ gedit hello.c
ubuntu@ubuntu:~$ gcc -o hello hello.c
ubuntu@ubuntu:~$ ./hello
Welcome to Open Source Lab
ubuntu@ubuntu:~$
```

4. Compile the source from any open source package

Installing ganache(open source package) using AppImage

```
ubuntu@ubuntu:~/Downloads$ sudo chmod a+x ganache-2.4.0-linux-x86_64.AppImage
[sudo] password for ubuntu:
ubuntu@ubuntu:~/Downloads$ sudo cp ganache-2.4.0-linux-x86_64.AppImage /usr/bin
ubuntu@ubuntu:~/Downloads$ ./ganache-2.4.0-linux-x86_64.AppImage
12:41:13.025 › Checking for update
12:41:13.378 › Generated new staging user ID: f8615ccc-cfcb-5663-93b5-6f96070461c5
12:41:16.956 › Update for version 2.4.0 is not available (latest version: 2.4.0, downgrade is disallowed).
```

Compiling an open source package

Results:

The compiling from the source is studied and executed.

18CS2064 - Open Source technologies Lab | URK17CS023