Ex 2 KERNAL CONFIGURATION, COMPILATION AND INSTALLATION
Date: 29.08.20

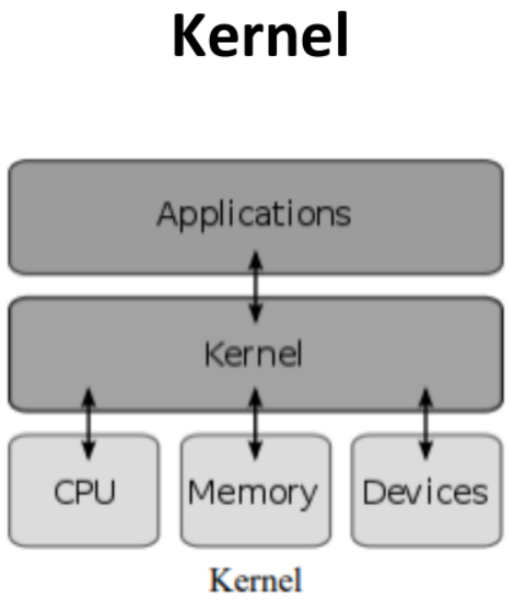
Aim:

To configure, compile and install linux kernel using source code.

Description:

Kernel is the main component of most computer operating systems; it is a bridge between applications and the actual data processing done at the hardware level.

The kernel's responsibilities include managing the system's resources (the communication between hardware and software components).



Kernel

Kernel usually provide methods for synchronization and communication between processes called inter-process communication (IPC).

The Linux kernel is the operating system kernel used by the Linux family of Unix-like operating systems. It is one of the most prominent examples of free and open source software.The Linux kernel is released under the GNU General Public License version 2 (GPLv2), and is developed by contributors worldwide.

Commands:

Sl. No.	Command Name	Meaning	options
1.	**Uname**	The **uname** command reports basic information about a computer's software and hardware.	**-a**, **--all** print all information, in the following order, except omit **-p** and **-i** if unknown **-s**, **--kernel-name** print the kernel name **-r** print the kernel release
2.	**Tar**	he **tar** command used to rip a collection of files and directories into highly compressed archive file commonly called tarball or **tar**, gzip and bzip in Linux	**-C,** --directory DIR **-f,** --file F **-j,** --bzip2 **-p,** --preserve-permissions **-v,** --verbose **-z,** --gzip
3.	**Ln**	make links between files	**-s**, **--symbolic:** make symbolic links instead of hard links **-f**, **--force** remove existing destination files **-i**, **--interactive** prompt whether to remove destinations
4.	**Make**	utility for building and maintaining groups of programs	**-b, -m** prints online help and exitThese options are ignored for compatibility with other versions of make.. **-d** Print debugging information in addition to normal processing.

5.	**Make clean**	Removes all previous configurations	---
6.	**Make install**	the make program takes the binaries from the previous step and copies them into some appropriate locations so that they can be accessed	---

Steps involved in configuration:

1. Build-essential package is required to be installed beforehand.

```
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$ sudo apt-get install build-esse
ntial libncurses-dev bison flex libssl-dev libelf-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.8ubuntu1).
The following additional packages will be installed:
  libfl-dev libfl2 libsigsegv2 m4 zlib1g-dev
Suggested packages:
```

2. **The version of the kernel can be identified using the uname command or found in /proc/version file**

```
jojo@jojo-VirtualBox:~$ uname -r
5.4.0-48-generic
```

```
jojo@jojo-VirtualBox:~$ cat /proc/version
Linux version 5.4.0-48-generic (buildd@lcy01-amd64-010) (gcc version 9.3.0
(Ubuntu 9.3.0-10ubuntu2)) #52-Ubuntu SMP Thu Sep 10 10:58:49 UTC 2020
```

3. Once kernel has been downloaded, move it to /usr/src

```
jojo@jojo-VirtualBox:/usr/src$ cd linux-5.8.10/
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$
```

4. **Unzip and move to the directory**

```
jojo@jojo-VirtualBox:~/Downloads$ sudo mv linux-5.8.10.tar.xz /usr/src
[sudo] password for jojo:
jojo@jojo-VirtualBox:~/Downloads$ cd /usr/src
jojo@jojo-VirtualBox:/usr/src$
```

5. **Now we create a symlinks to the kernel tree.**
   $ln -s /usr/src/linux-3.2 /usr/src/linux

```
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$ sudo ln -s /usr/src/linux-5.8.1
0 /usr/src/linux
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$
```

```
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$ sudo make mrproper
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$
```

6. Use make command to save .config file using ‘make config’ command.

```
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$ sudo make menuconfig
scripts/kconfig/mconf  Kconfig
#
# using defaults found in /boot/config-5.4.0-48-generic
#
/boot/config-5.4.0-48-generic:3815:warning: symbol value 'm' invalid for ISDN_CAPI
/boot/config-5.4.0-48-generic:8245:warning: symbol value 'm' invalid for ASHMEM
/boot/config-5.4.0-48-generic:9205:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
/boot/config-5.4.0-48-generic:9206:warning: symbol value 'm' invalid for ANDROID_BINDERFS
/boot/config-5.4.0-48-generic:9274:warning: symbol value 'm' invalid for INTERCONNECT


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

7. Use make clean command to clear up unnecessary files. Then use ‘make’ command.

```
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$ sudo make clean
jojo@jojo-VirtualBox:/usr/src/linux-5.8.10$ sudo make
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/conf.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTCC  scripts/kconfig/util.o
  HOSTLD  scripts/kconfig/conf
scripts/kconfig/conf  --syncconfig Kconfig
  SYSTBL  arch/x86/include/generated/asm/syscalls_32.h
  SYSHDR  arch/x86/include/generated/asm/unistd_32_ia32.h
  SYSHDR  arch/x86/include/generated/asm/unistd_64_x32.h
  SYSTBL  arch/x86/include/generated/asm/syscalls_64.h
  HYPERCALLS arch/x86/include/generated/asm/xen-hypercalls.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_32.h
  SYSHDR  arch/x86/include/generated/uapi/asm/unistd_64.h
```

8. After the compiling, make modules-install and install to set it up on /boot directory.

Results:

The linux kernel has been configured, compiled and executed.

Youtube Link:

https://youtu.be/oskDsHDQEQg