Page No.
Date 5/12/20

Case study on linux

1) Who does the developement ?

→ Linux, computer operating system created in the early 1990s by Finnish software engineer Linus Torvalds and the free softwar foundation (FSF). While still a student at the university of Helsinki, Torvalds started developing linux to create a system similar to MINIX, a UNIX operating system. Linux grew throughout the 1990s because of the efforts of hobbyist developers.

2) How is it organised ?

→ In order to make the most effective use of your linux system, you must undertand how linux organizes data. If you're familiar with microsoft windo or another operating system, you'll find it easy to learn how linux organizes, because most operating system organize data in rather simil ways. This section explains how linux organizes. The directories of a linux system are organized as a hierarchy. Unlike MS-DOS, which provides the seperate hierarchy for each partition.

C) How is it licensed?

→ Linus has placed the linux kernel under the GNU General Public License, which basically means that you may freely copy, change, and restrictions on further distribution and you must make the source data available. the licences of the utilities and programs which come with the installations vary. Much of the code is form the GNU project at the free software foundation, and is also under the GPL. some other major programs often included in Linux distribution are under a BSD license and other similar licenses.

A) How is the source code managed?

→ RCS manages single files at a time; CVS uses the RCS toolset to manage all of the source code for a project.. Linux uses essentially GIT to manage releases of new versions; updates to the central source tree are managed by a "benevolent despot" (Linus Torvalds and some "trusted lieutenants").

B) How are stable releases done?

→ At the end of the review cycle, the ACKed patches will be added to the latest - stable releases, and a new - stable release will happen. Security patches

will be accepted into the stable tree directly from the security kernel team, and not go through the normal review cycle.

6) How are stable releases done?

6) → What communication methods are used?

→ Linux supports three types of interprocess communication mechanisms that first appeard in Unix™ system V (1983). These are message queues, semaphores and shared memory. These system V IPC mechanisms all share common authentication methods.

There are some types of communication methodes :- 1> signals
2> pipes
3> sockets ① system V IPC machanism
② Message queues
③ semaphores
④ shared memory

7) How are bugs tracked?

→ What an hour of Googling bring out is that mainstream linux kernel bugs are being logged in two distinct systems

Mailing list :- This is the Linux kernel developement discussion and bug reporting mailing list.

The mailing list is archived by a number of services e.g. Gmane, Ikml, MARC, Mail archives, Indiana.

8) ~~What~~ How does it interact with other projects ?

→ The ~~s~~ projects can be implemented using Linux or windows OS. If anyone intrested to propose a new ~~video~~ idea, the preapar an abstract. The projects are devided into two parts ⇒ Hand on and research project

1) Hand on :- Each group must provide a poicerpoint presentation describing all the details of the project. if a group of student are planning to do a project, each individual students must clearly specify part in the project.

2) Research Projects :- All these projects must be done individually. students are expected to provide a powerpoint presentation as well as final paper.

9) What has the project dcumented about itself?

→ * History :- It has grown from a small number of c files under a license probibiling comercial distribution to the 4.15 version in 2018 with more than 23.3 million lines of source code not counting coments, under the GNU.

* coding style requirement :- functions should be short and sweet & do just one thing. They

should fit on one or two screenfuls of text and do one function is ~~interesty~~ inversly proportional to the completly & & indentation level of that function.

* code of conducts: The Linux code of conduct is bassed on the contributor convent, version 1.4. The contributor ~~corse~~ content has been adopted by hundreds of open-source projects. The contributors content was created by Coraline Ada Ehonke a software developer.
* continous integration requirements :- With 14,000 changesets per releases from other 1,700 different developer its clear that the linux kernal moves quickly & brings plenty of complexily.
kernal bugs range from small annayances to larger problems, such as system crashe & data loss.

10) What does it do?

→ Linux is an open source operating system. An operating system is the software that directly manages a system's hardware and resources, like CPU, memory and storage. The OS sits between applications and hardware and makes the connections between all of your software and the physical resources that do the work.

11) Who maintains it ?

→ Linus Torvalds, the creator of Linux, is the maintainer of Linux kernal developement. He is generally the last word on wheather code gets merged into the official Linux kernel or not. The Linux kernel has literally thousands of developers, but linus torvends has the final say.

12) How is the projecter structured ?

→ When install ALM, the installation program creates a project repository on the application servers file system. By default the project repository located under the application deployment directory. The project re-positary contains the sa and qc sub-folders. The sa directory stores global XML files, style sheets, tamplets and reports to be used by all projects in the project repository. The qc directory is a working area for a group of domains that are shared by multiple users.

13) How do the developers communicate ?

→ Mailing lists are the main communication channels in the Linux kernel. For newcomers that would like to learn more about the linux kernel developmen

there is the kernelnewbies resource and #kernelnewbies IRC channel on OFTC. This online resource provides information on basic kernel development questions.

iii) What has happened recently?

→ Find files that have been modified recently in linux

⇒ There are various occasions when we want to search for files that have been changed recently.

for ex :- as a system admin, we're responsible to maintain and configuration computer systems. Well explore the find utility which is the most common way to archive the intended purpose.

iv) How do I buid it?

→ Building a ~~cet~~ linux has its advantages and dis advantages. However, new Linux admin find it difficult to build linux. Building of linux needs to understand few things and then type a couple of commands.

The procedure to build the linux kernel.

1) Grab the latest kernel from kernel.org

2> verify kernel
3> untar the kernal tarball
4) copy existing linux kernel config file
5) build linux kernel.

16) How do I contribute?
→ one of the biggest -and the fastest moving -open source projects, the linux kernel is composed of about 53,600 files and nearly 20-million lines of code, with the more than 13,600 programmers contributing to the project world wide.

There are steps to contributing the kernel

1> prepare your system.
2> download the linux kernal code repository.
3> Build your kernel
4) Make a branch and switch to it.
5) Updata your kernel to point to the letest code base
6) Make a change to the code base
7) commit your changes and create a patch.