Install software

install from source code

- step 1: get code or download code
  - file format .tar and compress format .gz /.bz2 : GZIP / BZIP
- step 2: decompress and unarchive and untar
  - tar -xzvf <path/file name>
    - for tar and .gz
  - tar -xjvf <path/file name>
    - for tar and .bz2
- step 3: go to new directory mad — cd <directory path and name>
- step 4: read " Install " file located into directory
- step 5: configure installation policy — run ./configure or ./<script name>
  - file configure script and checking requirement
- step 6: compile code to binary
  - make
    - we need to move to the main package directory
  - gcc -v — directory: /usr
    - gnu c compiler information
- step 7: Install app — make install
- step 8: run app
  - type app name and use switches

install from package

- RPM package .rpm (red hat base) — install :
  - rpm -ivh <rpm path>
    - just install
  - rpm -Uvh <rpm path>
    - update and install
  - rpm -Fvh <rpm path>
    - just update
  - architecture is important : <app name and version> < architecture> .rpm // i386 :32bit //x86-64 :64bit and .....
- DEB package .deb (debian base)
- show info (query to package database)
  - rpm -qa | less
    - show all package in os
  - rpm -qa |grep -i <name>
    - search for app or package
  - rpm -e <app name>
    - erase package
  - rpm -qi <package name>
    - show all info about app
  - rpm -ql <package name> | less
    - show app directory and install location
  - rpm -qf <file-path>
    - show app version by give directory address
- download source for package
  - rpmfind.net // rpm.pbone.net

automatic install

- yum install <app name>
  - auto download and install if repository server has this app
- /etc/yum.conf
  - linux yum configuration file
- /etc/yum.reposd.d
  - yum repository location for change it
- yum repolist
  - showing repository servers for us and update xml database
- yum search < package name>
  - search for package in repository servers
- yum info <package name>
  - show package information in repository server

Library management

- libraries directory — /lib /lib64 /usr/lib
  - all libraries directory in linux
- ldd <program path / binary path>
  - show required library for some program
- ldconfig
  - rebuild cache file of library list or update info : /etc/ld.so.cache
- ldconfig -p | less
  - show human readable of ld.so.cache (binary)
- share object

Kernel module

- /lib/modules
  - linux kernel module location : module for install device file and driver for hardware
- lsmod
  - list currently loaded module
- modinfo <module name>
  - show info of installed module
- rmmod <module name>
  - remove or unload module from kernel
- insmod <module path>
  - install or load module to kernel
- modprobe <module name>
  - install or load module without directory path
- depmod
  - create cache file of modules : modules.dep
- kernel object of micro kernel