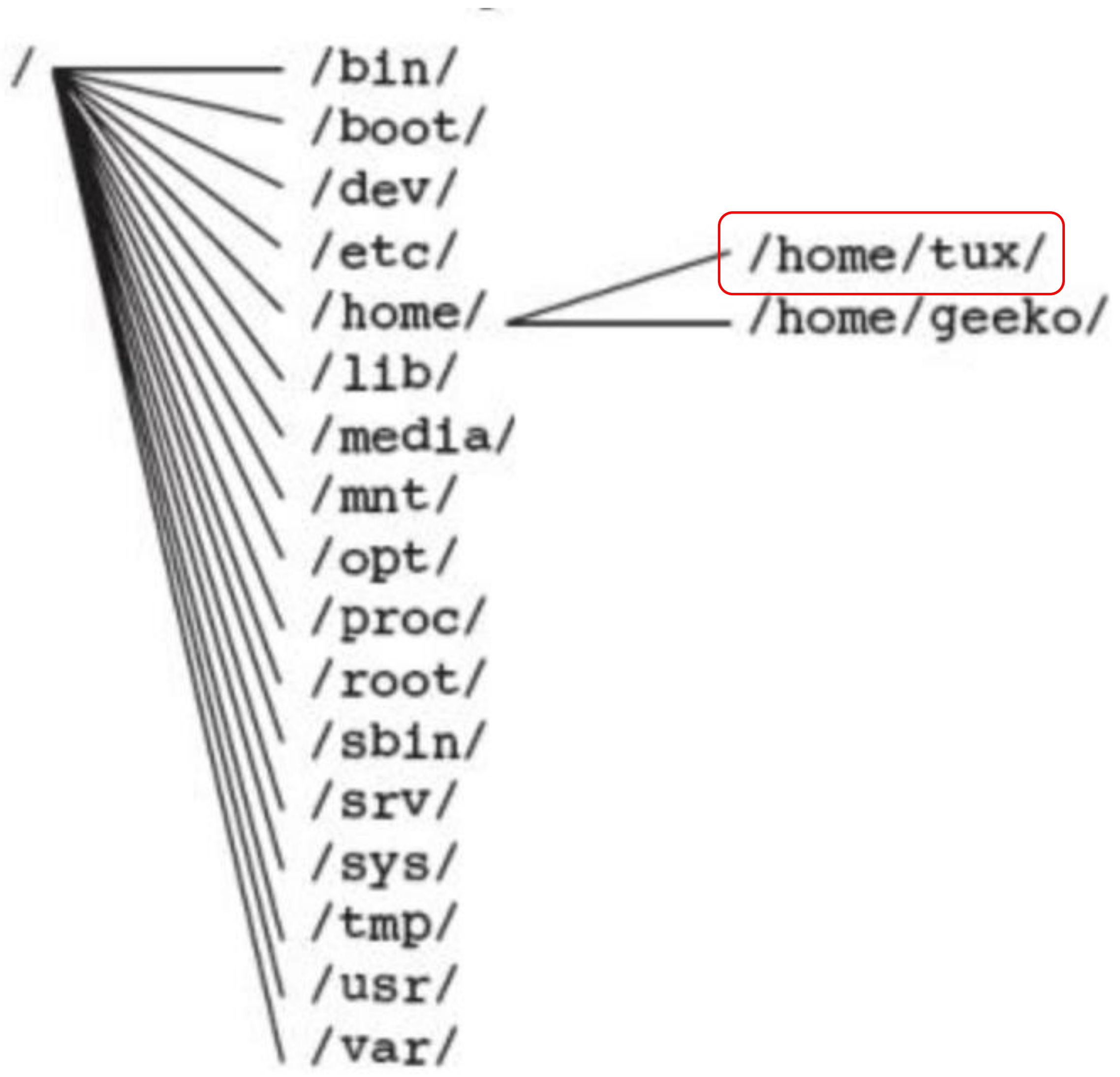
Linux Files and Directories: Filesystem Hierarchy Standard (FHS)

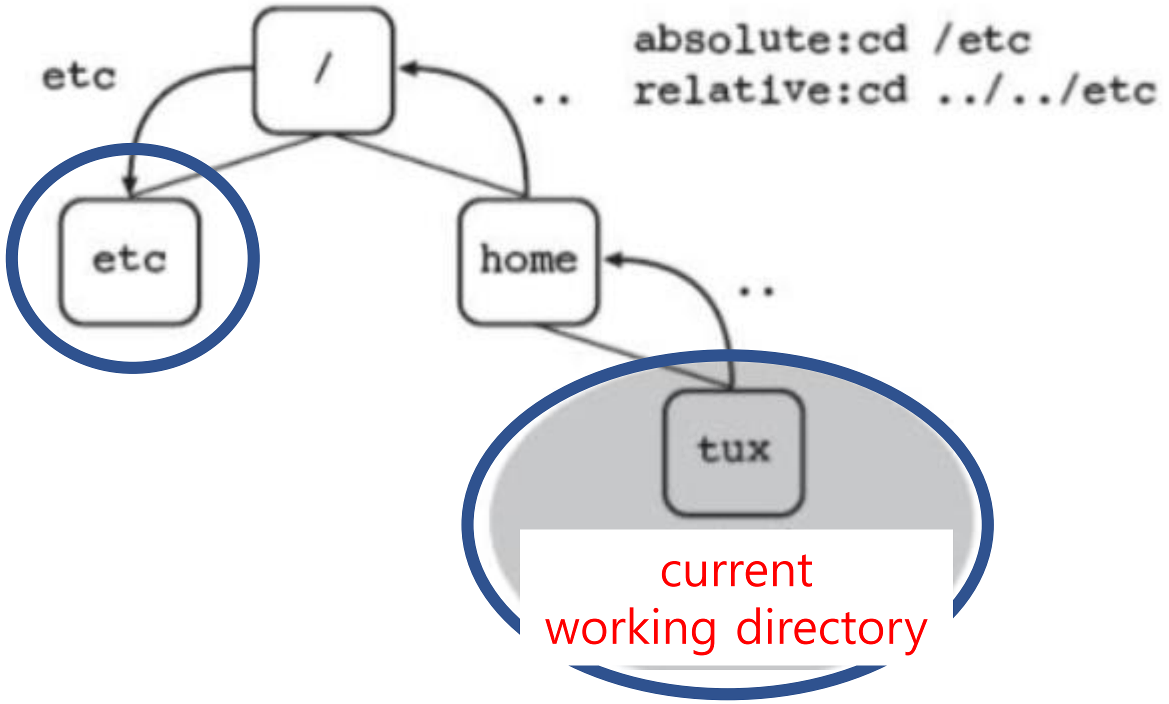
HGU

UNIX File System

- UNIX File System is Logical Method of Organizing and Storing Large Amount of Information in way that makes it easy to manage
- A file is a smallest unit in which the information is stored.
  - All data in UNIX Organized into files.
  - All files are organized into directories.
- Linux Files are stored in a single rooted, **hierarchical** file system
- Linux Places all the partitions under the root directory by ‘mounting’ them under specific directories.

The Hierarchical Structure of FHS





- relative path vs. absolute path

Understanding Linux File System Hierarchy (1)

- To understand the Linux file system, you need to know the following :
  - Root Directory (/)
  - Essential Binaries for Use by All users (/bin/)
  - Boot Directories (/boot/)
  - Device Files (/dev/)
  - Configuration Files (/etc/)
  - User Directories (/home/)
  - Libraries (/lib/)
  - Mount Points for Removable Media (/media/*)
  - Application Directory (/opt/)
  - Home Directory of the Administrator (/root/)
  - System Binaries (/sbin/)
  - Data Directories for Services (/srv/)

Understanding Linux File System Hierarchy (2)

- To understand the concept of the Linux file system,
    - Temporary Area (/tmp/)
    - user-space programs and files that are not essential for boot (/usr/)
    - Variable Files (/var/)
    - Process Files (/proc/)
    - System Information Directory (/sys/)
    - Mount Point for Temporarily Mounted File Systems (/mnt/)
    - Directories for Mounting Other File Systems

Root Directory (/)

- The root directory refers to the highest layer of the file system tree
- Only directories are located here, not files
- When the system is booted, the partition on which root directory is located is the first one mounted
- All programs that are run on the system start must be available on this partition
- The following directories always have to **be on the same partition** as the root directory:
  - /bin/, /sbin/, /lib/, /dev/, /root/, and /sys/
  - Essential for system boot, initialization, and emergency recovery

/bin/ : Essential Binaries for Use by all Users

- /bin/bash : the bash shell
- /bin/cat : display files
- /bin/cp : copy files
- /bin/dd : low-level copy files byte-wise
- /bin/gzip : compress files
- /bin/mount : Mount file systems
- .bin/rm : delete files
- /bin/vi : vi editor

Libraries (/lib/)

- contains **Essential Shared Libraries** that are required by the system for **booting** and running the core functionalities.
- These libraries are used by the binaries located in /bin and /sbin to function properly.
- Common Files in /lib
  - libc.so : C Libaray
  - libm.so : math library
  - Kernel modules: /lib/modules that are peace of codes loaded into kernel as needed to support hardware
- /lib64: contains libraries specifically for 64-bit binaries
- Libraries for user-installed applications are generally found in **/usr/lib** or **/usr/local/lib**, which are not critical for booting or system recovery.

Boot Directory (/boot/)

- /boot/ contains static files of the boot loader
  - These are files required for the boot process (with the exception of configuration files)
- The backed-up information for the Master Boot Record (MBR) and the system map files are also stored here
  - These contain information about where exactly the kernel is located on the partition
- This directory also contains the kernel
  - According to FHS, the kernel can also be located directly in the root directory

Configuration Files (/etc/)

File	Description
/etc/SuSE-release	version number of the installed SUSE Linux Enterprise Server
/etc/inittab	Configuration file for the init process
/etc/init.d/*	scripts for starting services
/etc/grub.conf	configuration file of GRUB
/etc/modprobe.conf	configuration file of kernel modules
/etc/DIR_COLORS	Specifies the colors of ls
/etc/X11/XF86Config	Configuration file of the X windows System
/etc/fstab	Table of the file systems automatically mounted at system start
/etc/passwd	User database; all information except passwords
/etc/shadow	Encrypted passwords of users
/etc/group	Database of user group
/etc/cups/*	Files for CUPS printing system
/etc/hosts	Allocation of computer names to IP address
/etc/motd	Welcome message adter a user logs in (message of the day)
/etc/issue	Linux Welcome message before the logic prompt

Device Files (/dev/)

- Each hardware component existing in the system is represented as a file in /dev/
- The hardware components are addressed via these files by writing or reading to or from one of the files.
- Two types of Device Files Exist:
  - Character special files (or character devices)
    - ‘talks’ to device character-by-character (1 byte at a time)
    - Examples: printer, virtual terminals, serial devices
  - Block special files (or block devices)
    - ‘talks’ to device 1 block at a time (1 block can be 512bytes to 312KB)
    - Examples: Hard disk, floppy disk, CD burners.

Device Files (/dev/, cont'd)

Device	Device Files	Description
Terminal	/dev/console /dev/tty1	The system console The first virtual console, reachable by pressing Ctrl+ALT+F1
Serial ports	/dev/ttyS0 /dev/ttyS*	the first serial port
Parallel ports	/dev/lp0 /dev/lp*	the first parallel port
Floppy disk drive	/dev/fd0 /dev/fd*	the first floppy disk drive. If the drives are addressed via the device file fd0 and fd1, the kernel tries to recognize the floppy disk format itself
IDE hard drive	/dev/hda /dev/hdc /dev/hd*	the first IDE hard drive on the first IDE controller the first IDE hard drive on the second IDE controller To label the partitions, the device names are given number. Number 1 to 4 refer to primary partitions, higher numbers to logical partitions. Example: /dev/hda1 is the first primary partition on the first IDE hard drive

Device Files (/dev/, cont'd)

Device	Device File	Description
IDE CD-ROM/DVD drive	/dev/hd*	The drives are named in the same way as the IDE hard drivers. This means that the CD-ROM/DVD drive /dev/hdd is the second drive on the second IDE controller
SCSI hard drives	/dev/sda /dev/sda*	the first SCSI hard drive With SCSI hard drives, the device names are given numbrs to label the various partitions. For example, /dev/sda1 is the first primary partition on the first SCSI hard drive
SCSI CD-ROM/DVD	/dev/scd0 /dev/scd*	The first SCSI CD-ROM/DVD drive

System Binary (/sbin/)

- Contains important system administration programs
- Programs in /sbin/ can also, as a rule, be run by normal users, but only to display configured values

File	Description
/sbin/conf.d/*	contains more scripts from SuSEconfig family; they are called up by /sbin/SuSEconfig
/sbin/yast	Administration tool for SUSE Linux Enterprise Server
/sbin/fdisk	Modifies partitions
/sbin/fsck	Checks file systems (file system check)
/sbin/init /sbin/mkfs	initializes the system creates a file system (formatting)
/sbin/shutdown	shut down the system

Regular User Directories (/home/)

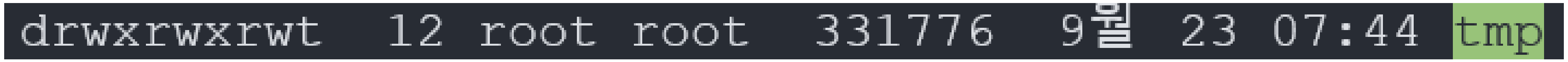
- Every user on a Linux system has his own area in which to create and remove files: its <u>home directory</u>
- Individual configuration files can be found in the user's home directory
  - .profile : user's private logic script
  - .bashrc : configuration file for bash
  - .bash_history : List of commands previously run in bash
- If there are no special settings, the home directories of all users are located beneath /home/
- The home directory of a user be addressed via "~"

Home Directory of the Administrator (/root/)

- The home directory of the system administrator is not located beneath /home/ like that of a normal user
- Preferably, it should be on the same partition as the root directory,"/"
  - Only then is it guaranteed that the user root can always log in without a problem and have her own configured environment available

Temporary Area (/tmp/)

- Directory used **for temporary files** that are created by programs, services, and users
- Not meant to persistent beyond a short period of time
  - **Auto cleaning** : cleared at boot or after a certain period of inactivity
  - **Volatile Storage**: /tmp is mounted as tmpfs meaning contents reside in RAM
- It is **world-writable**, meaning that **all users** can create and delete files in /tmp. To maintain security, the sticky bit is set on /tmp, meaning that although any user can create files in /tmp, they can only delete their own files.
  - **Sticky Bit** (represented by t in the file permissions) prevents users from deleting or modifying other users' files in /tmp. For example, the permissions for /tmp might look like this:



```
drwxrwxrwt  12 root root  331776  9월  23 07:44 tmp
```

The Hierarchy below /usr/ (**U**nix **S**ystem **R**esources)

- contains **user applications, utilities, and system libraries** that are <u>not essential for booting the system </u>but are necessary for normal multi-user operations.

Directory	Description
/usr/X11R6	FIles of X Window System
/usr/bin/	Almost all executable programs not essential for booting the Linux
/usr/lib/	Libraries
/usr/local/	Locally installed programs, now frequently found in the directory /opt/
/usr/sbin/	Programs for system administration
/usr/share/doc/	Documentation
/usr/share/man	The manual pages (command descriptions)
/usr/src/	SOurce files of all programs and the kernel (if installed)

Variable Files (/var/)

- Contains files that can be modified while the system is running
  - Data that changes
  - Growing data : log, spooled data, cache data
- Exmaple)
  - /var/spool : printer queue or mail queue waiting for later process
  - /var/cache : cached data (previously computed or downloaded file to reuse)
  - /var/tmp : temporary that need to persist between reboots
  - /var/log : log files generated by system and services (/var/log/syslog)
  - /var/lib : stores dynamic data related to the state of applications
  - /var/lock : lock files to prevent multiple concurrent processes to the same resource
- Persistent data: Unlike /tmp, which is cleared when rebooting, many files in /var remains across reboot

Data Directories for Services (/svc/)

- The directory /srv/ contains subdirectories filled with data of various services
- For example:
  - The files of the Apache web server are located in the directory /srv/www/
  - The FTP server files are located in the directory /srv/ftp/

System Information Directory (/sys/)

- The **/sys directory** is a **virtual filesystem** created and managed by the Linux kernel. It represents **system and hardware information**, including devices, drivers, and kernel settings.
  - Real-time System information and allows both reading hardware states and, writing to modify system behavior or configure devices.
- The files and directories within /sys are <u>not regular files</u> but are instead <u>interfaces to kernel data structures</u>, allowing user applications to interact with the system at a lower level.
  - **(Example 1)** You can adjust the *brightness of a laptop's backlight* by writing to /sys/class/backlight/acpi_video0/brightness.
  - **(Example 2)** /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq shows the current *CPU frequency.*
  - **(Example 3)** /sys/class/net/eth0/address: displays the *MAC address of the network* interface.

Mounting a File Systems

- **mounting** refers to the process of making a file system accessible at a certain point in the directory structure.
- A directory must exist at the point where you intend to mount the file system
  - This directory is referred to as **the mount point**
  - Mounted file system does <u>not have to be on a local</u> hard disk
- Use <u>mount</u> and <u>umount</u> commands
  - If you mount a file system to a non-empty directory, existing contents of directory will not be accessible
  - In most cases, only the **user root** can mount and unmount directories

Mount Point for Temporarily Mounted File Systems (/mnt/)

- Standard directory for integrating file systems
- It should only be used for <u>temporary purposes</u>
  - # mount /dev/hda7 /mnt
  - # umount /mnt
- To specify a specific file system, use the option -t
  - If the file system format is not supported by the kernel, the command is aborted
  - **mount -t** *<file_system_type> <device> <mount_point>*
    - <file_system_types> : *ext4* (4th Linux filesystem), *vfat* (FAT32 used for windows and USB), *ntfs* (Windows NTFS), *iso9660* (CDROM), *nfs* (Network file system), *tmpfs*
  - If you do not include any options with mount, the program tries out several file system formats

Directories for Mounting Other File Systems (cont'd)

- The directories that cannot be imported from other machines (machine-specific and directly tied to the local machine's hardware)

Directory	Description
/bin/	important programs
/boot/	kernel and boot files
/dev/	device files
/etc/	configuration files
/lib/	libraries
/sbin/	important programs for system administration

- Some of the directories that can be shared are:

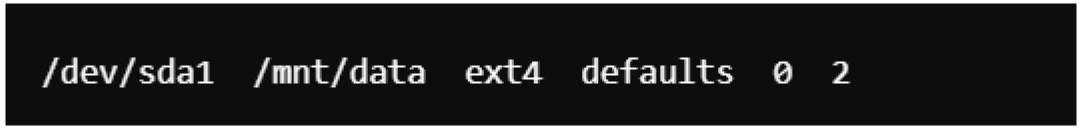
Directory	Description
/home/	Home directories
/opt/	Applications
/usr/	The hierarchy below /usr/

Mount Point for Removable Media (/media/)

- The **/media** directory is designated for mounting **removable media** like CDs, DVDs, USB drives, SD cards, and external hard drives
- On modern Linux systems, when a user inserts a USB drive or CD/DVD, the system typically automatically mounts the device under /media/, often creating a subdirectory with the name of the device or label:
  - /media/cdrom/
  - /media/usb/
  - /media/dvd/

Automated Mounting Filesystems

- Automated mounting when booting
  - use /etc/fstab
  - Example of /etc/fstab Entry



```
/dev/sda1  /mnt/data  ext4  defaults  0  2
```

    - /dev/sda1 : device to be mounted
    - /mnt/data : The directory of mount pont
    - ext4 : file system type
    - default: mount option (rw, suid, dev, exec, auto, nouser, async)
    - 0 : No dump backup
    - 2: Enable file system check (fsck) for non-root partition at boot

File Management Commands in Linux

File Types in the Linux System

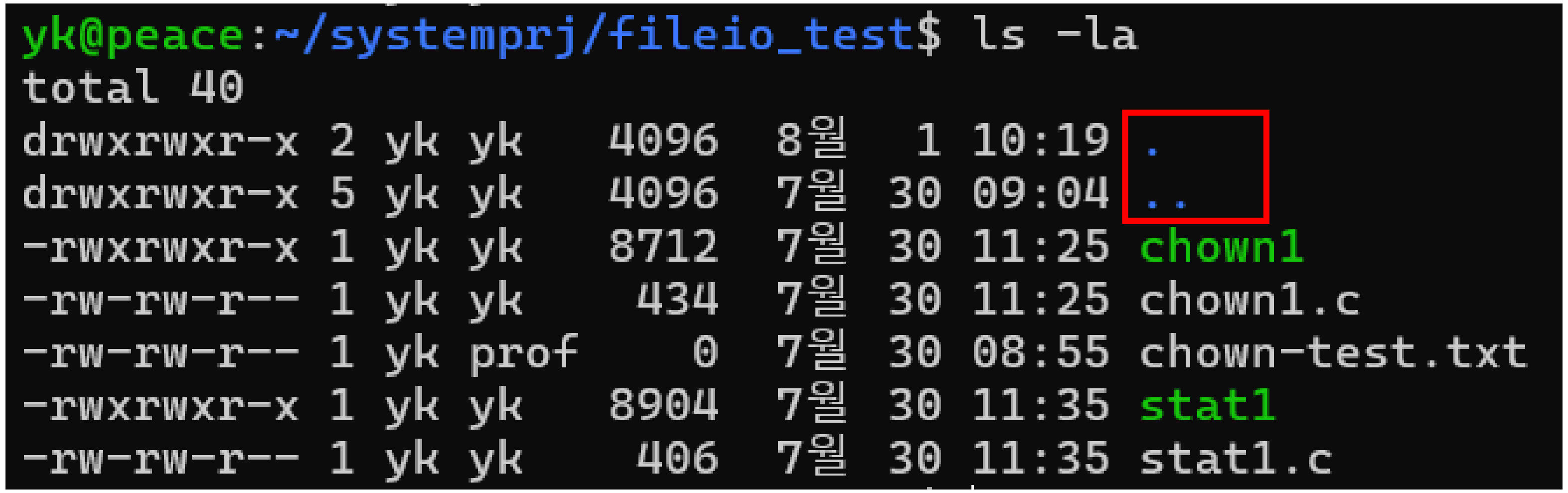
- The file types in Linux referred to as normal files and directories are also familiar to other operating systems
  - Normal Files
  - Directories
- Additional types of files are UNIX-specific
  - Device Files
  - Links
  - Sockets
  - Pipes and FIFOs

Normal Files

- **Normal files**: a set of contiguous data addressed with one name
  - This includes all the files normally expected under this term (such as ASCII texts, executable programs, or graphics files)
- You can use any names you want for these files—there is no division into filename and file type
  - A number of filenames still retain this structure, but these are requirements of the corresponding applications, such as a word processing program or a compiler

Directories

- **Directories** store both special and ordinary files. For users familiar with Windows or iOS, UNIX directories are equivalent to **folders**
  - two entries with which the structure of the hierarchical file system is implemented
    - One of these entries ("**.**") points to the directory itself
    - The other entry ("**..**") points to the entry one level higher in the hierarchy



```
yk@peace:~/systemprj/fileio_test$ ls -la
total 40
drwxrwxr-x 2 yk yk    4096  8월   1 10:19 .
drwxrwxr-x 5 yk yk    4096  7월  30 09:04 ..
-rwxrwxr-x 1 yk yk    8712  7월  30 11:25 chown1
-rw-rw-r-- 1 yk yk     434  7월  30 11:25 chown1.c
-rw-rw-r-- 1 yk prof     0  7월  30 08:55 chown-test.txt
-rwxrwxr-x 1 yk yk    8904  7월  30 11:35 stat1
-rw-rw-r-- 1 yk yk     406  7월  30 11:35 stat1.c
```

Device Files

- Each piece of hardware (with the exception of network cards) in a Linux system is represented by a **device file**
  - These files represent links between the *hardware components or the device drivers* in the kernel and the *applications*
- Every program that wants to access hardware must access it through the corresponding **device file**
  - The programs write to or read from a device file
  - The kernel then ensures that the data finds its way to the hardware or can be read from the file

Links

- **Links** are references to files *located at other points* in the file system
- Data maintenance is simplified through the use of such links
  - Changes only need to be made to the original file
  - The changes are then automatically valid for all links
- **Symbolic Link** : is used for referencing some other file of the file system. Symbolic link is also known as Soft link

Sockets, Pipes, and FIFOs

- A **Socket** refers to a special file with which data exchange between *two locally running processes (inter-process communication)* using FIFO
- **Pipes :** Unix allows you to link commands together using a pipe. The pipe acts as temporary file which only exists to hold data from one command until it is read by another. Pipe is a one-way flow of data.
- **FIFO** (first in first out) or **named pipe** is a term used for files used to exchange data between processes • The file can exchange data in one direction only

Change Directories and List Directory Contents

- The prompt of a shell terminal contains the current directory (such as tux@da10:~)
- The tilde (~) indicates that you are in the user's home directory
- Commands:
  - **cd** <path>: change directory to <path>
    - <path> can be '..' or '~'
    - cd : move from anywhere to home directory
  - **pwd** : print working directory
    - pwd –p : prints the physical directory without any symbolic link
  - **ls** : list
    - ls –a : display also hidden files (file name begin with .)
    - ls –l : detailed list (long list)
    - ls –R : recursive including subdirectories
    - ls –F : a character indicate file type (/ for directory, * for executable, @ for symbolic link)
    - ls –t : files are sorted by date of modification
    - ls –u : files are sorted by date last access

Create and View Files

- Create a New File with touch
  - **touch** <file> : changes the time stamp of a <file> or create a new <file> with a size of 0 byte.
- View a File with cat, more, less
  - **cat <file_name>** : type command in DOS
  - **less < file_name>** : display the contents of a file page by page quickly

keystroke in less	description
spacebar	move one screen down
b	move one screen up
Down-Arrow	move one line down
Up-Arrow	move one line up
/pattern	pattern search
n	move to the next instance in search for the pattern

View a File with head and tail

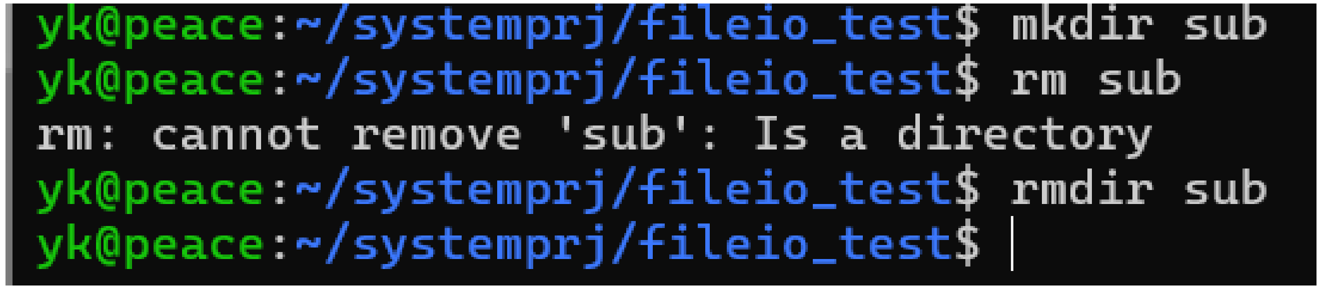
- Used to view the first or last lines of a file
  - **head** <file>
  - **tail** <file>
- By default, they show 10 lines
- head -20 displays the first twenty lines
- tail -f <file> displays a continuously updated view of the last lines of a file

Copy and Move Directories/ Files

- Copy Directories and Files: **cp** <source> <target>
  - **cp –r** <source dir> <tagert dir> : copy all the contents of <source dir> to new <target_dir>
  - **cp –r** <source dir>/* <existing_dir> : copies all the contents of <source directory> except hidden files to <existing_dir> recursively.
- Move Directories and Files: **mv** <source> < target>
  - <source> can be a directory or a file
  - it renames <source > to <taget>; copy <source> to <target> and delete <source>
  - if <target> is an existing directory, the <source> is copied under <target>

Create and Delete Directories/ Files

- Create directory: mkdir <directory>
- Delete files and directories: rm <path>
  - **rmdir** <directory> : delete empty directory
  - **rm –r** <directory> : remove directories and their contents recursively



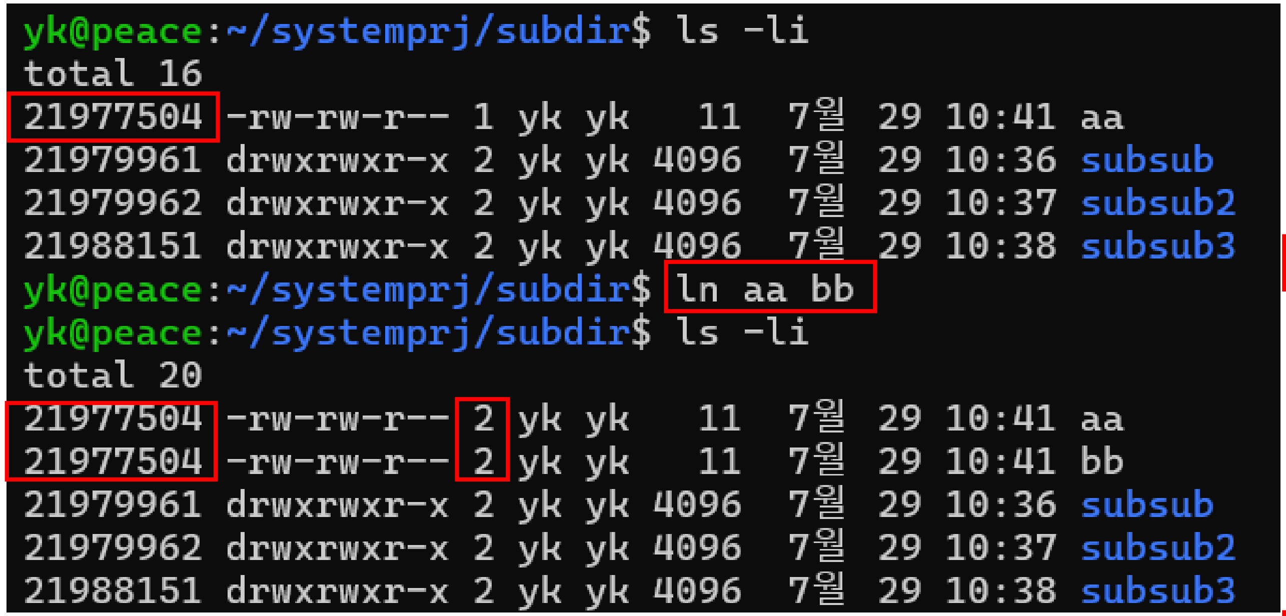
```
yk@peace:~/systemprj/fileio_test$ mkdir sub
yk@peace:~/systemprj/fileio_test$ rm sub
rm: cannot remove 'sub': Is a directory
yk@peace:~/systemprj/fileio_test$ rmdir sub
yk@peace:~/systemprj/fileio_test$ 
```

Link Files

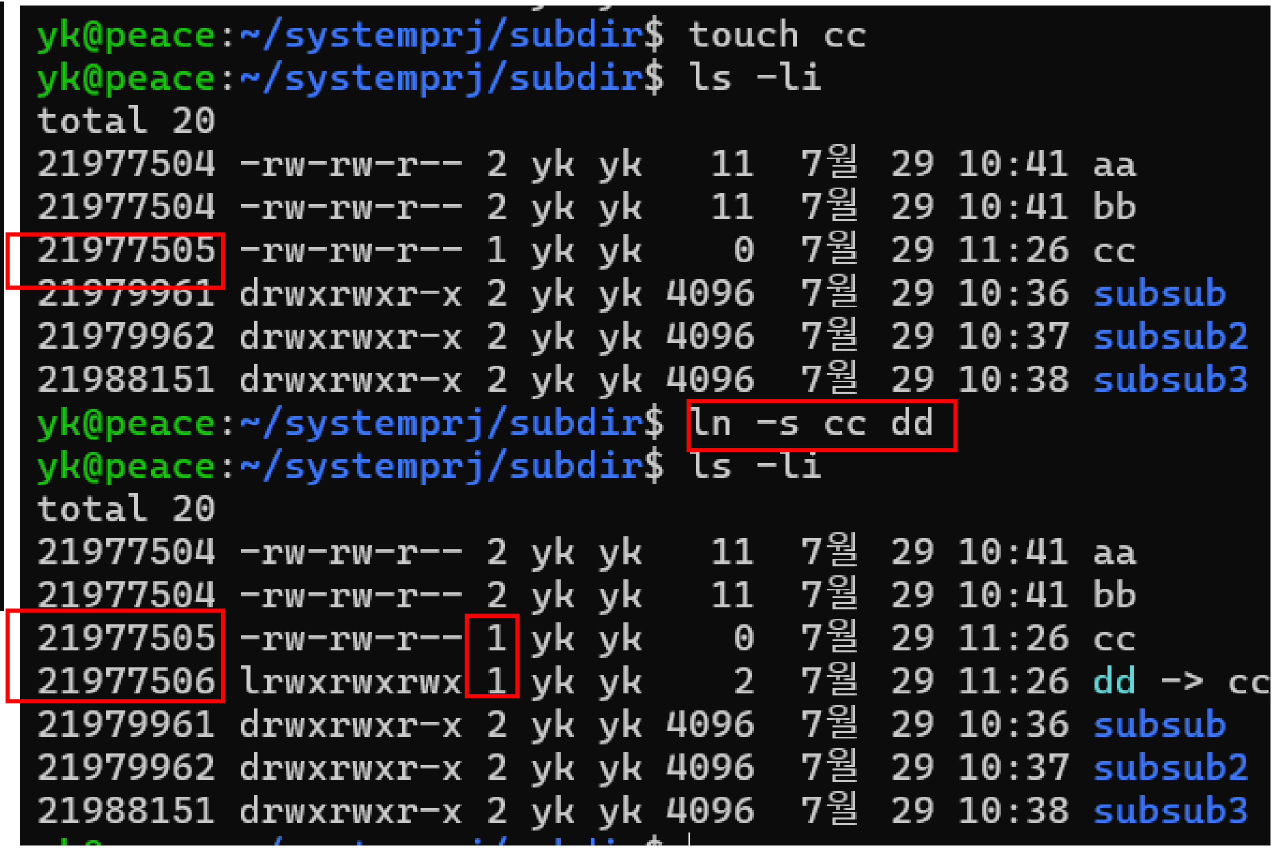
- Each file is described by an *inode*
  - To see the inode number you an enter **ls –i**
  - Each inode has a size of 128 bytes
  - An inode contains all the information about the file besides filename
- Link: a reference to a file
  - Create a <u>Hard Link </u>using **ln**, which points to the inode of an already existing file
    - hard links can only be used when both file and the link are in the same file system
  - Create a <u>Symbolic Link </u>using **ln –s**; a symbolic link is assigned its own inode
    - Original and Link has different inode
    - Using symbolic Link, you can create link to directories *(Hard Link* is **not allowed for directory**)

Link Example

- make link with ln command
- check with ls –i command



```
yk@peace:~/systemprj/subdir$ ls -li
total 16
21977504 -rw-rw-r-- 1 yk yk   11  7월  29 10:41 aa
21979961 drwxrwxr-x 2 yk yk 4096  7월  29 10:36 subsub
21979962 drwxrwxr-x 2 yk yk 4096  7월  29 10:37 subsub2
21988151 drwxrwxr-x 2 yk yk 4096  7월  29 10:38 subsub3
yk@peace:~/systemprj/subdir$ ln aa bb
yk@peace:~/systemprj/subdir$ ls -li
total 20
21977504 -rw-rw-r-- 2 yk yk   11  7월  29 10:41 aa
21977504 -rw-rw-r-- 2 yk yk   11  7월  29 10:41 bb
21979961 drwxrwxr-x 2 yk yk 4096  7월  29 10:36 subsub
21979962 drwxrwxr-x 2 yk yk 4096  7월  29 10:37 subsub2
21988151 drwxrwxr-x 2 yk yk 4096  7월  29 10:38 subsub3
```



```
yk@peace:~/systemprj/subdir$ touch cc
yk@peace:~/systemprj/subdir$ ls -li
total 20
21977504 -rw-rw-r-- 2 yk yk   11  7월  29 10:41 aa
21977504 -rw-rw-r-- 2 yk yk   11  7월  29 10:41 bb
21977505 -rw-rw-r-- 1 yk yk    0  7월  29 11:26 cc
21979961 drwxrwxr-x 2 yk yk 4096  7월  29 10:36 subsub
21979962 drwxrwxr-x 2 yk yk 4096  7월  29 10:37 subsub2
21988151 drwxrwxr-x 2 yk yk 4096  7월  29 10:38 subsub3
yk@peace:~/systemprj/subdir$ ln -s cc dd
yk@peace:~/systemprj/subdir$ ls -li
total 20
21977504 -rw-rw-r-- 2 yk yk   11  7월  29 10:41 aa
21977504 -rw-rw-r-- 2 yk yk   11  7월  29 10:41 bb
21977505 -rw-rw-r-- 1 yk yk    0  7월  29 11:26 cc
21977506 lrwxrwxrwx 1 yk yk    2  7월  29 11:26 dd -> cc
21979961 drwxrwxr-x 2 yk yk 4096  7월  29 10:36 subsub
21979962 drwxrwxr-x 2 yk yk 4096  7월  29 10:37 subsub2
21988151 drwxrwxr-x 2 yk yk 4096  7월  29 10:38 subsub3
```

Find Files

- find <path> -name <patterns>
  - find / -name game : looks for a file named ‘game’ starting / directory
  - find /home –user joe : find every files under the directory /home owned by joe
  - find /usr –name *stat : find every files under the directory /usr ending ‘stat’
- locate <pattern>
  - alternative to find –name
  - locate <pattern>
  - package findutils-locate must be installed
  - searches through database previously created (/var/lib/located), making it faster (database is daily updated automatically or manually by updated)
- whereis <command>:
  - returns binary (-b), sourcecode (-s) or manual (-m) for the specified command
- which <command>
  - searches all the paths listed in the variable PATH for the specified command and returns the full path of specified command