WHAT IS INODE (index) number of a file -> is like a “finger-print”, unique for each file on the Linux file system.

i-node is an index (data structure) that provides information about the file such as if the file is a directory or regular file, etc.

running ls -i will display the i-node of files/directories

Hard links

* Reference to the same index (physical data) on a file system
* Creates a file that shares the same i-node number with original file
* Good way to back up your file, as long as one hard link remains, the data is preserved.
* Change original file, changes all other files that are hard-linked
* Disadvantage is that they take-up extra space, and cannot hard link directories (only files).
* Cannot hard link files from other unix/linux servers

Symbolic (soft) links

* Indirect pointer to a file. Contains the pathname to the original file
* Like a shortcut to other files. Can create symbolic links on different unix/linux servers, and also for directories
* Not good for backup purposes, if you delete the original file, data is lost and the symbolic link will point to a nonexistent file (broken link)

Managing processes

All commands/programs (tasks) that are running on Unix/Linux system are referred to as processes.

* Each process has an owner
* Each process has a unique ID (PID)
* Processes keep their PID for their entire life
* Usually a parent sleeps when a child is running
* Hierarchical

Commands used to manage processes

ps -> displays snapshot information about processes, basic listing of processes in current user’s terminal, such as PID, process names

ps -l -> detailed listing in current user’s terminal, such as PID, PPID (parent PID), status

ps -ef -> detailed listing ALL processes running on entire system

ps aux -> detailed listing of processes for ALL users and background running services

ps -U username -> basic listing of processes running for a particular user

top -> provides a realtime status of running processes, ctrl-c to exit

ctrl-c -> terminates a process running in the foreground

ctrl-z -> sends a process running in the foreground to the background

fg -> moves a background job from the current environment into the foreground, fg %job-number

bg -> runs the most recent process that was placed into the background, bg %job-number

jobs -> displays the status of jobs

kill -> kills the process(job), kill PID, kill -9 PID, kill %job-number, kill -9 %job-number

Foreground vs background processes

* Command & - will run the command in the background

ALIASES

* Is a nickname to an existing command or group of commands
* Lasts only for the current session, it will be lost when you log out from the session
* Can make it permanent by adding it to .bashrc file

Command history

.bash\_history in your home directory stores recently executed command lines

fc -l -> displays last 16 commands

history | more -> display all stored commands

!num -> re-execute an issued command number by command number

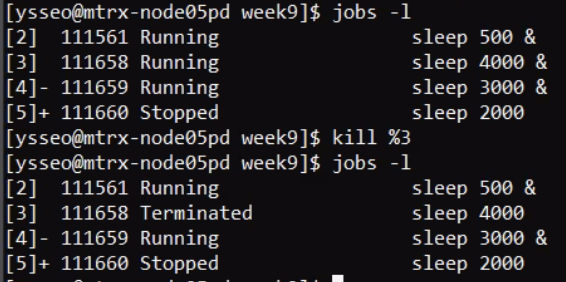
!xxx -> re-run a most recent previously-issued command beginning with string “xxx”

Text

Description automatically generatedText

Description automatically generated

kill %3: kill 3rd process



kill PID

A black screen with white text

Description automatically generated with low confidence

bg %1 : bring the 1th process back to foreground

Graphical user interface

Description automatically generated with medium confidence

ctrl – z bring the process to background

Text

Description automatically generated

broken link

A picture containing calendar

Description automatically generated

symbolic link (content sync)

Text

Description automatically generated

hard link (create the link)A picture containing text

Description automatically generated

hard link (content sync)

Text

Description automatically generated