Saturday, March 26, 2022 10:32 AM

System ----- can be accessed by different users

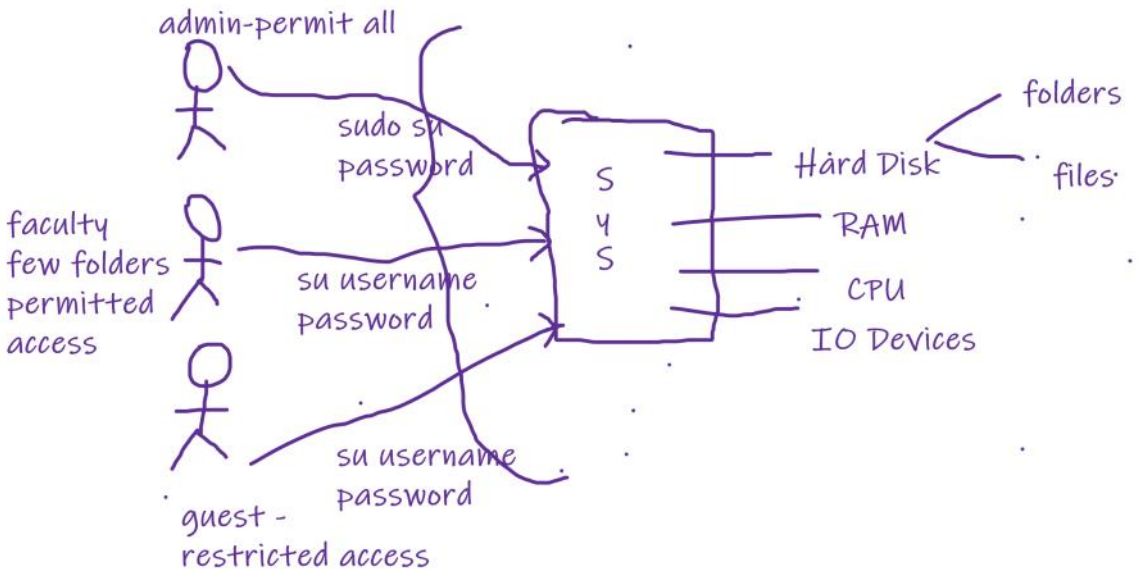
Authorization ----which user can access which files and folders
Users are AUTHORIZED differently

A1 user	Is authorized to access(read +write) all folders
A2,A3,A4,A5 user	Is authorized to access(read +write) only his or her home folder If I am logged in as blue , I can make changes ONLY in /home/blue If I am logged in as yellow , I can make changes ONLY in /home/yellow I may only see the content of blue not change If I am logged in as cdacos, I can make changes ONLY in /home/cdacos
A6	Is authorized for only reading a particular folder

Super User sudo su	Authorization for creating users , changing passwords, Accessing all folders for read and write ,……
Non super users Blue, yellow, cdac	They cannot create users or change passwords --not authorized They can fully access their own home folders /home/blue etc
su username password ????	

Password is used to AUTHENTICATE (to ensure that the user is real)

USERS are required for AUTHENTICATION and AUTHORIZATION !!!!



Ex1 --- we open the ubuntu, login with suitable user name ,
In the ~ folder , create a new directory called as cprograms

In cprograms we create a file hello.c
We compile the file using gcc
We run the file

cat	command is used to show the contents of a file cat ./cprograms/hello.c
ls	Command is used to show the contents of a folder

Compile	$ gcc -o hello hello.c
Command	Gcc
Option	-o value
Value	Hello ---- this is the name of the output file
If -o is not given then the name of the output file is by default is a.out	
hello.c	is the name of the input file

If gcc command is not available then download the package using	sudo su apt-get update apt install gcc

---

Process Creation ------ every process gets a process ID

To see the PID of all processes ----------

Windows	GUI Task manager
Linux	CLI command ps -e
Command	ps (process status)
Option	-e (for entire linux on the system)
ps without any option	It shows the processes active on current TERMINAL

TTY	terminal type
For example we have opened three terminals	pts/0, pts/1,pts/2

Every command in linux is a PROCESS	**cd , pwd , su , mkdir, rmdir , rm -r , vi , cat ,ls ,logout**

Every Terminal in linux is also a process of a SHELL program

SHELL Program = CLI terminal program
This programs allows the user to enter command and see output
Default shell program is called as BASH
We can find this program in **/bin** folder
Other shells that can be installed --- Ksh, CSH , TCSH , ...
Every shell has its own set of commands .We are focusing on BASH commands !!!

Linux system has a bootstrap process process-id 1 ----- init process

The init process will produce child processes , which will create more child processes and this goes on.
The process of creating child processes is called as FORK !!!!!!

Except the init with pid=1 , all other processes have a parent processes !!!

TO see the parent process

ps - ef

UID	User that started the process
Pid	Process id
PPID	Parent process id
Command	The name of the process
…	

Ex2 --Modify the C program ---- add infinite loop in the program

from another terminal observe the pid and ppid of our hello program using **ps -ef** command

Later terminate the program using ctrl-c and check again using ps -ef (no longer seen)

Run the program again and terminate it using a Kill

kill pid command

kill 214

System calls ----------

Program---------->system call---------->System

What is a system call ? A function call made to a kernel function

We will use 2 system calls ----- getpid() getppid()

Ex3 ---- Write a c program test.c in the cprogram folder inside the ~

Print your name and print the return values of getpid and getppid

---

Fork = activity of creating a child process from the parent process .

fork is a system call .

WHAT HAPPENS IN FORK ?

Every process gets a process space. Parts of the process space code, data, stack, heap

Fork CREATES A COPY of the process space …… duplicating the calling process

The copy is the CHILD process and the original is the parent process

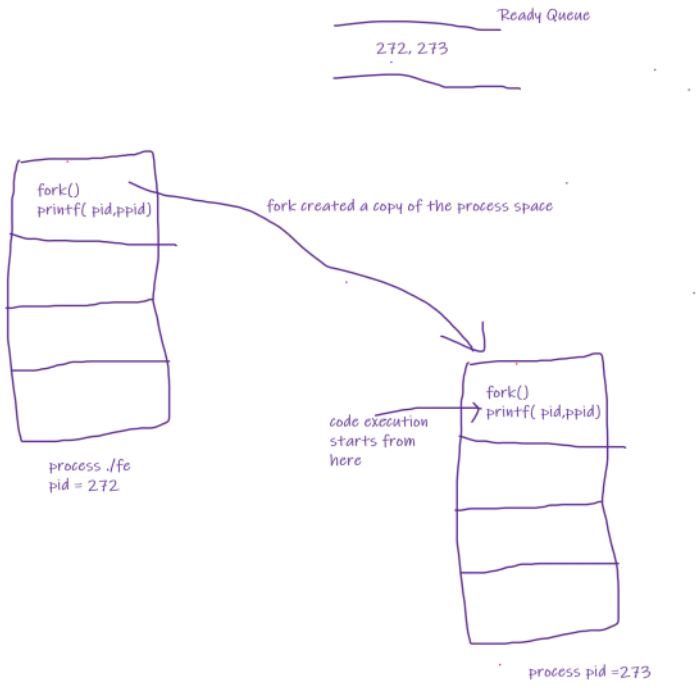
The child process gets a unique pid , runs separately and competes with parent process in RR .

Ex4 ---- write a c program forkex.c in the cprogram folder inside the ~ .

59 ------by starting terminal

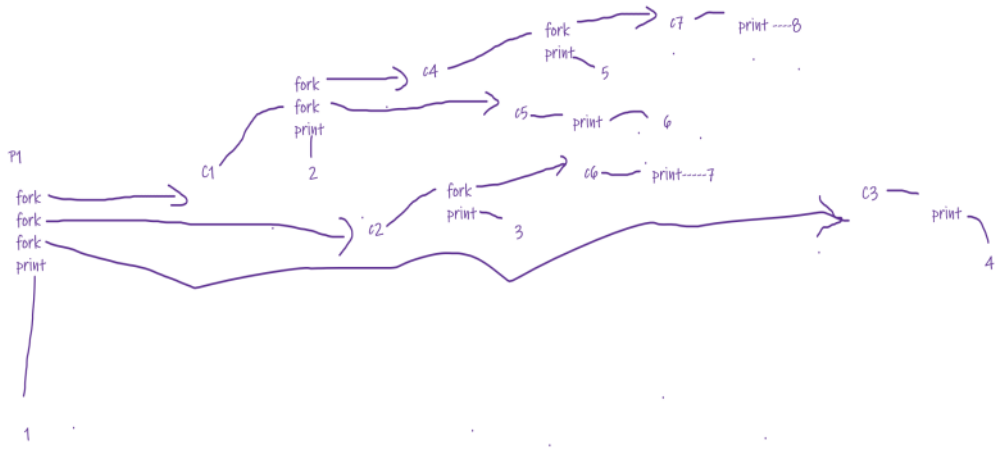
----- 272 ./fe

----------273 fork



Ex 5 ---

Write a c program forkex2.c in ~/cprograms
Add 3 calls to the fork()



If you have consecutive fork() then the formula is **2 raised to number of fork() calls = number of processes created**

$2^3 = 8$

EX6 --- Try out the ps , ps -e, ps -ef commands , kill command as discussed in class

---