

Punjab University College of Information Technology
University of the Punjab
BCSF15 (Term Fall 2017)

CMP-325 Operating Systems

Student ID: _____

Quiz 07

Time: 30 mins

Marks: 20

Quiz will be cancelled if any student is found looking at his/her neighbor's paper or using any unfair mean

1. Briefly explain proc filesystem.

[2]

Window into the running linux kernel:

- ⇒ `man proc` # very comprehensive page.
- ⇒ `proc` file-system is used to view what the linux kernel and all processes are doing currently. It contains many files.
- ⇒ Files in `proc` are only on RAM not in HDD.
- ⇒ size of all files in it is zero.
- ⇒ They are magic objects that are created and deleted on the fly.
- ⇒ They provide access to data structures, parameters and statistics of the running kernel and processes.
- ⇒ We can change configuration on kernel by writing to some of the files in `proc` filesystem.
- ⇒ It contains 1-directory for each running process named according to PID.

2. Write and describe file in `proc` directory containing OS version info.

[2]

⇒ `/proc/version` file contains string describing kernel OS type and kernel version. Also contains data it was compiled and built - and also compiler version.

⇒ different commands/processes uses info in this directory. e.g:

`uname`

3. Write name of some files present in directory with PID of SHELL.

[2]

⇒ Shell `proc` directory contains:

- `cmdline`: contains passed cmd-line arguments
- `environ`: contains shell env variables.
- `limits`: contains different limits (Y-cd: name, soft-limit, hard limit, unit)
- `stat`: process status info. Also used by `ps`. (numeric form)

4. Briefly explain the use of /proc/[PID]/task directory. [2]

task: Every proc dir contains it. It contains all contents of process dir. This is for mgmt of threads. For each ~~pro~~ thread within this process (1878), kernel provides a sub-dir named /proc/PID/task/TID

5. What is the use of /proc/self. [2]

/proc/self: symbolic soft-link to pid of process that called self. Used by processes to get info about themselves.

6. Differentiate between 'at' and 'batch'. [2]

⇒ "at" vs "batch"

- ⇒ in "at" we need to specify a specific time, at which task is performed.
- ⇒ in "batch" we don't specify time. Task/Command is executed whenever CPU load drops below 80%. and on standalone PC our CPU load is normally below 80%. so command will be executed within a minute or even earlier.

7. Differentiate between 'cron' and 'anacron'. [2]

⇒ cron vs anacron

minimum granularity: cron → 1min (tasks can be scheduled for every minute)
anacron → 1day
↳ normally for root

- ⇒ for cron system should be in running state at scheduled time so 24/7 uptime.
- ⇒ for anacron if system was down at scheduled time then it execute tasks whenever the system turns on. (even after months) missed

8.

[2]

a. Command to view running daemons.

ps -ajx

- a: for viewing processes of all users
- j: to view additional info like session ID of controlling terminal
- x: show processes that don't have controlling terminal

b. Command to start *cron* service.

sudo service cron start

9. Write cron entries for following:

[2]

a. To perform some task at 1:25 and 1:50 on 15th of every month.

25,50 1 15 * * task

b. To execute a command every minute on Wednesday

* * * * 3 command

10. Explain the use of *cron.deny* and *cron.allow* files.

[2]

/etc/cron.deny # users who can't use crontab
/etc/cron.allow # users who can run cron jobs and use crontab

→ if *cron.allow* and *cron.deny* both don't exist then only "root" can use crontab.
→ if both files exist then *cron.deny* is ignored