Lab 08 – Automation of System Tasks

Illustrate automation of basic tasks like monitoring memory consumption, check connectivity, etc., at different frequencies.

Sometimes, user may have tasks that need to be performed on a regular basis or at certain predefined intervals. Such tasks include backing up databases, updating the system, performing periodic reboots and so on. Such tasks in linux are referred to as **cron jobs (Crontab).** Cron jobs are used for **automation of tasks** that come in handy and help in simplifying the execution of repetitive and sometimes everyday tasks.

Commands to Schedule Tasks:

cron:

- The **cron** is a software utility, offered by a Linux-like operating system that automates thescheduled task at a predetermined time.
- It is a **daemon process**, which runs as a background process and performs the specified operations at the predefined time when a certain event or condition is triggered withoutthe intervention of a user.
- The **crontab** (abbreviation for "cron table") is list of commands to execute the scheduledtasks at specific time. It allows the user to add, remove or modify the scheduled tasks.
- The crontab command syntax has **six fields** separated by space where the **first five** represent the **time to run the task** and **the last one is for the command**.
  - Minute (holds a value between 0-59)
  - Hour (holds value between 0-23)
  - Day of Month (holds value between 1-31)
  - Month of the year (holds a value between 1-12 or Jan-Dec, the first three letters of themonth's name shall be used)
  - Day of the week (holds a value between 0-6 or Sun-Sat, here also first three letters ofthe day shall be used)
  - Command (**6^{th} Field**)

The rules which govern the format of date and time field as follows:

- When any of the first five fields are set to an asterisk(*), it stands for all the values of thefield. For instance, to execute a command daily, we can put an asterisk(*) in the week's field.
- One can also use a range of numbers, separated with a hyphen(-) in the time and date field to include more than one contiguous value but not all the values of the field. Forexample, we can use the 7-10 to run a command from July to October.
- The comma (,) operator is used to include a list of numbers which may or may not be consecutive. For example, "1, 3, 5" in the weeks' field signifies execution of a command every Monday, Wednesday, and Friday.
- A slash character(/) is included to skip given number of values. For instance, "*/4" in thehour's field specifies 'every 4 hours' which is equivalent to 0, 4, 8, 12, 16, 20.

Permitting users to run cron jobs:

- The user must be listed in this file to be able to run cron jobs if the file exists.
    - **/etc/cron.allow**
- If the cron.allow file doesn't exist but the cron.deny file exists, then a user must not belisted in this file to be able to run the cron job.
    - **/etc/cron.deny**
- **Note:** If neither of these files exists then only the superuser(system administrator) will beallowed to use a given command.

Simple Example 1:

Simple Example:

File Name: task.sh path: **/home/cbkpc/**(say for example)

Shell script:

```
echo    Welcome to Task Scheduler Demo
echo    Creating file on desktop
touch /home/cbkpc/Desktop/file1.txt
```

- Change the execution permission to task.sh
  chmod +x task.sh
- Run the commands in root
  - Su
  - Nano crontab –e

In the crontab file add the following line to execute command at particular time(forexample at 3:45 pm

  - 45 15 * * * /usr/bin/sh /home/cbkpc/task.sh
- Save and Exit. And Wait till 3.45pm
- After that task.sh will be automatically executed and new file1 will be created on Desktop

Simple Example 2: Monitoring Memory Consumption and remote server connectivity Checking

File Name: task2.sh path: **: /home/cbkpc /**(say for example)

Shell script:

```
echo Memory Consumption output is
in memoryoutput.txtfree >
/home/admincs/memoryoutput.txt
echo Checking Connectivity output is
in requestreply.txt ping –c 4
```

- Change the execution permission to task.sh
  - chmod +x task.sh
- Run the commands in root
  - Su
  - crontab –e

In the crontab file add the following line to execute command at particular time(for example at3:45 pm

  - 45 15 * * * /usr/bin/sh /home/cbkpc/task2.sh
- Save and Exit. And Wait till 3.45pm
- After that, Check the output by the command
  - cat /home/admincs/memoryoutput.txt
  - cat /home/admincs/requestreply.txt
- Alternatively /var/log/syslog can be verified about the execution of the Task.

Extra Example 1:

- Run /home/folder/gfg-code.sh every hour, from 9:00 AM to 6:00 PM, everyday.
    - **00 09-18 * * * /home/folder/gfg-code.sh**
- Run /usr/local/bin/backup at 11:30 PM, every weekday.
    - **30 23 * * Mon, Tue, Wed, Thu, Fri /usr/local/bin/backup**
- Run sample-command.sh at 07:30, 09:30, 13:30 and 15:30.
    - **30 07, 09, 13, 15 * * * sample-command.sh**

Creating cron jobs

- To create or edit a cron job as the root user, run the command
    - **# crontab -e**
- To create a cron job or schedule a task as another user, use the syntax
    - **# crontab -u username -e**
- For instance, to run a cron job as user Pradeep, issue the command:
    - **# crontab -u Pradeep -e**
- If there is no preexisting crontab file, then user will get a blank text document. Ifacrontab file was existing, The -e option allows to edit the file,

Listing crontab files

- To view the cron jobs that have been created, simply pass the -l option as shown
    - **# crontab**

-lDeleting a crontab file

- To delete a cron file, simply run **crontab -e** and delete or the line of the cron job thatuserwants and save the file.
- To remove all cron jobs, run the command:
    - **# crontab -r**

=== * ===