UNIVERSIDAD DEL BÍO-BÍO

Informe Tarea 2

Agendar Procesos Sistemas Operativos

Nombre:	Benjamín Martínez Jeldres
Carrera:	IECI
Profesor:	Ennio Pereira
Fecha:	11/06/2022
Asignatura:	Sistemas Operativos

Introducción

El objetivo de este trabajo es aprender e implementar una forma productiva de agendar procesos en Linux usando **Putty**, considerando agendar como programar la ejecución de una tarea para que se realice en una fecha posterior. Se usará el comando **Crontab** para que podamos agendar un proceso en una fecha determinada.

Desarrollo

Para comenzar daré la descripción y aplicación de **Crontab**.

Descripción: es un archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado por el usuario, este verificará la hora y la fecha en que se deberá ejecutar el comando, en este caso la hora será del servidor de **Putty.**

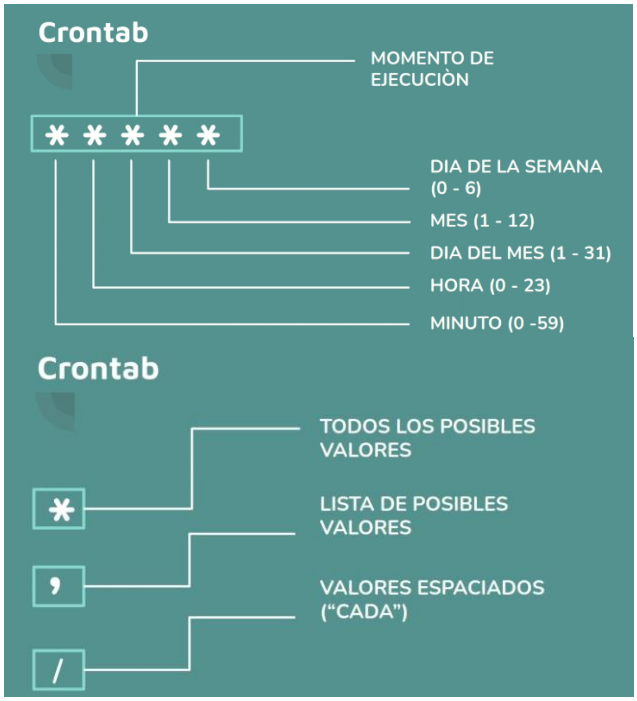
Aplicación: e implementado **Crontab** como una alarma en este trabajo, esta alarma será aplicada en mi horario académico de la Universidad, donde me avisará 10 minutos antes a través de un mensaje guardado en un archivo de texto, y mostrando la hora actual de ese momento, el mensaje dirá **“En diez minutos tienes clases, la hora es:”**.

El nombre del archivo de texto es **alarma.log,** allí se ira guardando el ultimo mensaje de la alarma programada y sobrescribiendo el mensaje anterior.

Para este trabajo tuve que hacer uso de **Python 3**, ya que **Putty** no me mostraba ningún resultado al usar Python.

Lo primero que hice fue crear un pequeño código en **crontab.py** a través de **Putty** donde importé la biblioteca **datetime**. Para poder guardar el mensaje tuve que usar **crontab -e** que se utiliza para entrar en **Crontab** y poder configurar las alarmas que necesito.

Tendremos 5 instrucciones:



Configuré 5 alarmas en **Crontab** las cuales se ejecutarían con **crontab.py,** luego asigné que **crontab.py** guardase el resultado en un nuevo archivo llamado **alarma.log.**

Algo muy importante al modificar **Crontab**, es que siempre hay que darle permisos de ejecución usando el comando **chmod +x nombre_archivo** y listo, solo queda esperar la fecha establecida para ver los resultados.

Horario Académico en el cual base la alarma:

Pasos a seguir en Putty :

```
bmartinez@loki:~$ ls
alarma.log  cancion.txt  chico.zip  crontab.py  datos.txt  foto.jpg  matrix
```

```
bmartinez@loki:~$ nano crontab.py
```

```
import datetime

print ("En 10 minutos tienes clases, la hora es:   " + str(datetime.datetime.now( ) ) )
```

```
bmartinez@loki:~$ crontab -e
```

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

00 08 * * 1,2,3  python3 /home/ssoo2022/bmartinez/crontab.py  >  /home/ssoo2022/bmartinez/alarma.log
30 12 * * 1   python3 /home/ssoo2022/bmartinez/crontab.py  >  /home/ssoo2022/bmartinez/alarma.log
00 11 * * 3   python3 /home/ssoo2022/bmartinez/crontab.py  >  /home/ssoo2022/bmartinez/alarma.log
00 17 * * 1,2,3,4  python3 /home/ssoo2022/bmartinez/crontab.py  >  /home/ssoo2022/bmartinez/alarma.log
30 18 * * 2,4  python3 /home/ssoo2022/bmartinez/crontab.py  >  /home/ssoo2022/bmartinez/alarma.log
```

```
bmartinez@loki:~$ chmod +x crontab.py
```

```
bmartinez@loki:~$ nano alarma.log
```

```
En 10 minutos tienes clases, la hora es:   2022-06-09 17:00:26.742740
```

Conclusión

Para finalizar, quiero decir que me costo bastante en algunos momentos, sobre todo el porque no me funcionaba al principio el código, ya que la mayoría de información está en inglés, tuve mucho que leer, hasta que vi un ejemplo en el cual se usaba **Python 3** y no **Phyton** así que lo modifiqué, y me funcionó, luego lo demás lo encontré mas sencillo, por lo que me pareció una tarea entretenida el configurar las horas de los horarios.
Siento que aprendí bastante en este trabajo, en relación a **Crontab,** también sobre la librería o biblioteca **datetime** y demás relacionados.