WUOLAH



sesion-4-ejercicios.pdf Practicas Móduloī

- 2° Sistemas Operativos
- Grado en Ingeniería Informática
- Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación Universidad de Granada

-50€ OFF

¡Viaje sorpresa en camper con tus amigos!

Descubre tu destino 2 días antes

Código: WAYNABOXSTUDENT





Instalar el at

[root@localhost ~]# mount none /mnt -t hostfs -o /home/usuario/Escritorio/pilar/segundo/so/practicas [root@localhost ~]# ls /mnt //Se monta todo lo que hay en la carpeta Ejercicios sesi??n1.odt ej3.2 Fedora14-x86-root_fs.gz kernel32-3.0.4.gz kernel32-3.0.rar P2.odt Portafolios_Practicas_SO_JesusGarciaGodoy.pdf paquetes SO-P-ManualEditor-vi.pdf paquetes.zip SO-P-Todos_GuiaPr??ctcas.pdf practica1, silvia.odt SO-P-Todos_MaterialModulo2.zip scriptarranque Sesion 2.odt seesion3.odt at-3.1.12-5.fc14.i686.rpm sesion3,modulo1,so.pdf [root@localhost ~]# rpm -ivh --nodeps at-3.1.12-5.fc14.i686.rpm error: open of at-3.1.12-5.fc14.i686.rpm failed: No such file or directory [root@localhost ~]# rpm -ivh --nodeps /mnt/at-3.1.12-5.fc14.i686.rpm [root@localhost ~]# service atd status atd is stopped [root@localhost ~]# service atd start //Si se para el atd. Mensaje: Can't open /var/run/ atd.pid to signal atd. No atd running? [OK] atd: [OK] <Control +D> para cerrar at Ejercicio 2vi genera-apunte #!/bin/bash ls -R /home >> ./listahome-`date +%Y-%j-%T-\$\$`



```
"genera-apunte" [New] 2L, 57C written [root@localhost ~]# at -f ./genera-apunte now + 1 minute job 3 at Wed Oct 18 01:48:00 2017
```

Tenemos que decirle a at donde queremos crear el archivo, en este caso ./, sino lo crea en su directorio, y debe tener permisos para acceder a ese directorio.

Si redirijo la salida dentro del propio atd se crea el archivo en el directorio donde ejecutamos la orden:

```
at now + 1 minute
at> ls -l ~>nuevo.txt
at> <EOT>
job 3 at Tue Oct 17 18:42:00 2017
[root@localhost ~]# ls -lR
.:
total 4
-rw-r--r-- 1 root root 56 Oct 17 18:42 nuevo.txt
Ejercicio 3-
[root@localhost ~]# at midnight
at> ls -l
at> <EOT>
job 4 at Wed Oct 18 00:00:00 2017
[root@localhost ~]# at midnight + 1 minute
at> ls -l
at> <EOT>
job 5 at Wed Oct 18 00:01:00 2017
[root@localhost ~]# at 17:30 tomorrow
at> ls -l
at> <EOT>
job 6 at Wed Oct 18 17:30:00 2017
[root@localhost ~]# at Dec 25
                                     //Si ponemos solo la fecha se ejecuta ese día a la misma hora en que estamos
at> ls -l
at> <EOT>
job 7 at Mon Dec 25 19:01:00 2017
[root@localhost ~]# at 00:00 Jan 1 //Si no ponemos año se ejecuta en el año que estamos
at> ls -l
at> <EOT>
iob 8 at Mon Jan 1 00:00:00 2018
[root@localhost ~]# at 00:00 Jan 1 2017
at: refusing to create job destined in the past
```

Lista de órdenes:

[root@localhost ~]# atq 6 Wed Oct 18 17:30:00 2017 a root 8 Mon Jan 1 00:00:00 2018 a root 7 Mon Dec 25 19:01:00 2017 a root





Descarga la APP de Wuolah. Ya disponible para el móvil y la tablet.







Continúa do



405416_arts_esce ues2016juny.pdf

Top de tu gi











- 5 Wed Oct 18 00:01:00 2017 a root
- 4 Wed Oct 18 00:00:00 2017 a root

Borramos el trabajo 5:

[root@localhost ~]# atrm 5 [root@localhost ~]# atq

- Wed Oct 18 17:30:00 2017 a root 6 8 Mon Jan 1 00:00:00 2018 a root
- 7 Mon Dec 25 19:01:00 2017 a root
- 4 Wed Oct 18 00:00:00 2017 a root

Para borrar todos los trabajos:

[root@localhost ~]# atrm 6 8 7 4 [root@localhost ~]# atq

Ejercicio 5-

vi prueba #!/bin/bash

nombrearchivo=`date +%Y-%j-%T` ps -ef > \$nombrearchivo echo Mi pid = \$\$ >> \$nombrearchivo

```
"prueba" [New] 5L, 101C written
[root@localhost ~]# chmod 777 ./prueba
[root@localhost ~]# at -f ./prueba now + 1 minutes
job 16 at Tue Oct 17 19:33:00 2017
[root@localhost ~]# ls
2017-290-19:33:00 prueba
[root@localhost ~]# cat 2017-290-19:33:00
        PID PPID C STIME TTY
                                        TIME CMD
UID
       1243
               1 0 18:40 ?
                               00:00:00 /usr/sbin/atd
root
       1276
              1 0 19:01?
                               00:00:00 /usr/sbin/anacron -s
root
                               00:00:00 [flush-98:0]
       1288
              2 0 19:16?
root
       1362 1243 0 19:32 ?
                                00:00:00 /usr/sbin/atd
root
root
       1363 1362 0 19:32 ?
                                00:00:00 sh
       1365 1363 0 19:32 ?
                                00:00:00 /bin/bash
root
root
       1367 1365 0 19:32 ?
                                00:00:00 ps -ef
Mi pid = 1365
```

El proceso padre de nuestro proceso, como podemos ver por el PPID, es **sh** que a su vez es hijo de atd (/**usr/sbin/atd**), es decir, atd lanza la ejecución del terminal sh el cual ejecuta nuestro script.

```
Ejercicio 6-
vi prueba2.sh
#!/bin/bash
find.-atime-1
"prueba2.sh" [New] 2L, 30C written
[root@localhost ~]# ls
2017-290-19:33:00 prueba prueba2.sh
[root@localhost ~]# chmod 777 prueba2.sh
[root@localhost ~]# at -f ./prueba2.sh now + 1 day
job 21 at Wed Oct 18 20:18:00 2017
[root@localhost ~]# at -f ./prueba2.sh 1>> ~/modificados_`date +%Y-%j-%T` now + 1 minute
job 24 at Tue Oct 17 20:21:00 2017
[root@localhost ~]# ls
2017-290-19:33:00 modificados_2017-290-20:20:42 prueba2.sh
modificados
                prueba
```

[root@localhost \sim]# at -f ./prueba2.sh 1>> \sim /modificados_`date +%Y-%j-%T` 2> \sim /errores now + 1 day

//No sé si funciona redirigiendo la salida aquí. Si le damos la orden directamente a atd y no a través de un script sí funciona redirigir la salida.

De esta forma funciona seguro:



```
#!/bin/bash
find . -atime -1 >> modificados_`date +%Y-%j-%T`
"prueba2.sh" 2L, 62C written
[root@localhost ~]# at -f ./prueba2.sh 2> ~/errores now (+ 1 day)
[root@localhost ~]# ls
errores modificados_2017-290-20:36:35 prueba prueba2.sh
[root@localhost ~]# cat modificados_2017-290-20:36:35
./modificados 2017-290-20:36:35
./errores
./prueba2.sh
./prueba
Ejercicio 7-
Ejercicio 9-
adduser pilar
su pilar
          //pasarnos a un usuario que no sea el root
vi prueba
#!/bin/bash
nombrearchivo=`date +%Y-%j-%T`
ps -ef > $nombrearchivo
echo Mi pid = $$ >> $nombrearchivo
crontab -e
* * * * * /home/pilar/prueba //poner la ruta absoluta del script
```

Ejecutamos crontab en el directorio home (no hace falta que sea el home salvo que no especifiquemos la ruta absoluta en el archivo de crontab y lo hagamos con ./) del usuario en segundo plano para no tener la terminal ocupada:

crontab & [pilar@localhost ~]\$ ls 2017-295-19:16:01 2017-295-19:18:01 prueba 2017-295-19:17:01 2017-295-19:19:01 [pilar@localhost ~]\$ cat 2017-295-19:19:01



WAYNABOXSTUDENT

?waynabox

Descubre tu destino 2 días antes

```
Código:
                                 www.waynabox.com
```

```
UID
        PID PPID C STIME TTY
                                        TIME CMD
root
       1105
              1 0 18:24 ?
                               00:00:00 /sbin/rsyslogd -c 4
       1130
              1 0 18:24 ?
                               00:00:00 /usr/sbin/sshd
root
        1156
                1 0 18:24 ?
                                 00:00:00 sendmail: Queue runner@01:00:00 for
smmsp
/var/spool/clientmqueue
               1 0 18:24 ?
                               00:00:00 crond
       1167
root
                               00:00:00 login -- root
       1179
               1 0 18:24 ?
root
root
       1180 1179 0 18:24 tty0
                                 00:00:00 -bash
root
       1208
               1 0 19:01 ?
                               00:00:00 /usr/sbin/anacron -s
                               00:00:00 [flush-98:0]
       1211
              2 0 19:01 ?
root
       1232 1180 0 19:13 tty0
                                 00:00:00 su pilar
root
                                 00:00:00 bash
       1233 1232 0 19:13 tty0
pilar
                                 00:00:00 CROND
       1273 1167 0 19:19?
pilar
       1274 1273 0 19:19 ?
                                 00:00:00 /bin/sh -c /home/pilar/prueba
       1276 1274 0 19:19?
                                 00:00:00 ps -ef
pilar
Mi Pid = 1274
Como vemos el proceso padre de mi proceso es /bin/sh -c /home/pilar/prueba, que a su vez tiene
como padre el proceso CRON.
[pilar@localhost ~]$ crontab -r //borrar el crontab que hemos creado
[pilar@localhost ~]$ crontab -l
no crontab for pilar
Ejericio 10-
su pilar
vi script1
#!/bin/bash
rm -v /tmp/varios/core* >>/tmp/listacores
chmod 777 ./script1
crontab -e
```

Una vez editado el archivo crontab no hace falta ejecutar crontab, se ejecuta sólo al haber editado el archivo con -e. Si creamos un archivo separado sí hay que ejecutar crontab <archivo>.

ls /tmp varios [pilar@localhost ~]\$ ls /tmp listacores varios [pilar@localhost ~]\$ cat /tmp/listacores removed `/tmp/varios/core1' removed `/tmp/varios/core2' removed `/tmp/varios/core4' [pilar@localhost ~]\$ ls /tmp/varios chr

* * * * * /home/pilar/script1



Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad

```
Ejercicio 11-
vi reducelista
#!/bin/bash
head -n 10 /tmp/listacores > /tmp/temporal
rm /tmp/listacores
mv /tmp/temporal /tmp/listacores
chmod 777 reducelista
mkdir /tmp/varios
cd /tmp
touch ./varios/core{1,2,6,7,3,45,t,d,f,t,j,6,9,12} ./varios/{hola,c,tu,you}
crontab -e
* * * * * /home/pilar/script1
* * * * * /home/pilar/reducelista
cat listacores
removed `/tmp/varios/core1'
removed `/tmp/varios/core12'
removed `/tmp/varios/core2'
removed `/tmp/varios/core3'
removed `/tmp/varios/core45'
removed `/tmp/varios/core6'
removed `/tmp/varios/core7'
removed `/tmp/varios/core9'
removed `/tmp/varios/cored'
removed `/tmp/varios/coref'
ls varios
c cr hola t tu you
```

Ejercicio 12-

