Taller Sistemas Operativos 10 a Noviembre 2017

10.1 - procesos ps, bash & [1] [2]

proceso : *programa ejecutándose* (**Importante**)

```
  cd ; cd tso/01/mio
ps
ps -l     | tee  ps.sal.antes-1               # 100
  bash
  ps -l   | tee  ps.sal.en-1                  # 120
  exit
ps -l     | tee  ps.sal.despues-1             # 140
```

Observamos los procesos, y su relación entre ellos:
`PID PPID` (*identificador de proceso* e *identificador de proceso padre*)

Una sesión anidada ha dado lugar a un proceso más en *el grupo* .

```
echo 'echo entro; ps -l ; echo salgo' > prps02
cat prps02
ps -l        | tee  ps.sal.antes-2            # 200
bash prps02  | tee  ps.sal.en-2
ps -l        | tee  ps.sal.despues-2          # 220
```

Observamos igualmente ... .

Un *escript* (programa del *intérprete de comandos*) **también** ha dado lugar a un proceso más.

```
date ; sleep 10 ; date
date ; sleep 10 & date                       # 250
```

La primera línea separa las fechas 10 u 11 segundos.
La segunda línea escribe las dos fechas sin esperar.

```
ps -l        | tee  ps.sal.antes-3            # 280
sleep 120 &
ps -l        | tee  ps.sal.despues-3          # 290
```

Ahora tenemos también un proceso añadido, pero con otro *esquema* (grafo).

10.2 - bash - variables [3]

```
echo $v1 $v2 $v3
v1=hola
v2=adios
v3=hasta la vista
v3="hasta la vista"                          # 350
echo $v1 $v3 $v2
export v4="nos vemos"                        # 360
bash
  echo $v1
  echo $v4
  export v5=voy                              # 390
```

[1] apuntes SSAA, cap. 16, pag. 126, 130-132
[2] apuntes SSAA, cap. 16, pag. 137, 141
[3] apuntes SSAA, cap. 22, pag. 229,230

```
  v6=vengo
  echo $v6 $v6
exit
echo $v5 $v6
echo $v1 $v1
v=saludos
echo ${v}1 "===" $v1                              # 440
```

Las variables exportadas las *heredan* los procesos hijos.

10.3 - bash - read [4]

```
  cd ; cd ../progs/bashes
echo read v1 v2 v3                      > pr10a
echo echo '$v1 "=" $v2 "-" $v3 "="'  >> pr10a
cat pr10a
bash pr10a
 uno dos tres
bash pr10a
 uno dos                                          # 500
bash pr10a
 uno dos tres cuatro cinco
```

La lista de *piezas* (*tokens*) leidos se asigna con flexibilidad a las variables de `read` .

10.4 - tar [5]

```
  cd ; cd tso/01
ls -R progs
du progs
tar cvf progs.tar progs
tar czvf progs.tgz progs                         # 550
gzip progs.tar
ls -l progs.*
```

Podemos observar los objetos bajo `progs`, su tamaño, ocupación, como se agrupan en `progs.tar.gz` y `progs.tgz` .

```
  rm progs.tar.gz
  cd mio
tar tzvf ../progs.tgz
tar xzvf ../progs.tgz                            # 600
ls -R progs
mkdir dt; cd dt
tar xzvf ../../progs.tgz                         # 610
ls -R
cd ..
ls -R
rm -fr progs dt/progs                            # 650   Cuidado !!
```

Vemos como se re-crea el *arbol/grafo* que se guardó con `tar c..f` .

---

[4] apuntes SSAA, cap. 22, pag. 245

[5] apuntes SSAA, cap. 20, pag. 213, 217