

Para hablar de pipes tenemos que comprender el concepto de subshell. Una subshell es un subproceso creado por la shell (o un shell script) que está en ejecución. La misma va a ser otra instancia de bash que, luego del fork, se va a convertir en un comando específico (por ejemplo, pstree) desencadenado por un `execv`.

Imaginen un árbol de procesos:

```
terminalVirtual (o sesión ssh por ejemplo)
      |
      --> bash (shell en ejecución)
```

Ahora, ¿para qué esto?. Bueno, para ejecutar comandos.

Esto va a suceder dependiendo de las circunstancias, es decir, de que tipo de comando estemos ejecutando (interno o externo), si utilizamos pipes, si ejecutamos en background (&), si indicamos explícitamente la creación de una subshell (con `()`) o si utilizamos sustitución de comandos:

Tipos de comandos:

Internos: no siempre crea una subshell. Depende de como se ejecute el comando. Por ejemplo: `# echo "so"`.

Externos: siempre va a generar una subshell a menos que se lo ejecute con `"exec"`, lo que reemplazaría la shell actual.

Algunas otras circunstancias posibles indistintas al tipo de comando que estemos ejecutando:

Uso de pipes: genera una subshell por cada termino de pipe.

Uso de modo background (&): genera una subshell.

Uso de sustitución de comandos: genera una subshell.

Ejemplo de árbol de procesos:

Si ejecutamos el siguiente conjunto de comandos:

```
# cat /etc/passwd | grep $USER
```

Vamos a tener lo siguiente:

```
terminalVirtual
      |
      --> bash --> cat
                        |
                        --> grep
```

Variables internas de BASH:

- `$$`: mantiene el PID de la shell (bash actualmente corriendo).
- `$PPID`: mantiene el PID del padre de la shell (terminal virtual, sesión ssh).
- `$BASHPID`: mantiene el PID de la shell / subshell actual. Puede coincidir con `$$` si estamos a la misma altura. A medida que se vayan creando subshells, las mismas van a tener su propio PID y van a diferenciarse de `$$`. Disponibles a partir de un fork.
- `$BASH_SUBSHELL`: mantiene el nivel de anidamiento de la subshell. Se va incrementando a medida que las subshell se van spawnedando (luego del `execv`).