

## Práctica 6: Automatización: preparación

- 1) Añadir 500 usuarios: user000, user002, user003 ... user499 con *passwords*: qwerty000 ... qwerty499, en cada uno de los S.O. de las maquinas virtuales de las prácticas anteriores.
  - ▶ Todos los usuarios deben estar perfectamente operativos con su directorio *home* y contraseña adecuadamente configurados.
    - ▶ es decir, **todos los usuarios deben poder ENTRAR EN EL SISTEMA tanto en modo texto como en el login gráfico, en el caso de que se use.**
  - ▶ Utilizar la política por defecto del S.O. en cuanto a los grupos primarios de los usuarios
  - ▶ Los usuarios creados utilizarán el *bash* como shell por defecto
  - ▶ Puede añadirse, si se considera necesario, ficheros al directorio */etc/skel* (o equivalente)

## Práctica 6: Automatización

Crear un shell script que crea copias de seguridad de los directorios de los usuarios.

- ▶ El script será ejecutado por el shell `/bin/sh`
- ▶ El script definirá una variable `BACKUP_DIR` con el directorio donde se almacenan las copias de seguridad (en el ejemplo `/var/backups`)
- ▶ El script definirá una variable `COMPRESOR` con el nombre del compresor que se usará para comprimir las copias de seguridad (en el ejemplo `bzip2`)
- ▶ A efectos de copias de seguridad se considerarán usuarios validos aquellos cuyo uid es mayor o igual que el primer usuario creado en la máquina (*usuario*)

```
#!/bin/sh
# autor: Donald Trump Trump
# autor: Bladimir Putin Rodriguez
BACKUP_DIR=/var/backups
COMPRESOR=/bin/bzip2
MINUID='cat /etc/passwd | grep ^usuario | cut -f3 -d:'
.....
```

## Práctica 6: Automatización

- ▶ El script incluirá como comentario los nombres de los autores
- ▶ Se hará copia de seguridad documental de los usuarios cuyo directorio *HOME* contenga el fichero '.backupDAY' o '.backupWEEK' (que indicaría copia diaria o semanal)
- ▶ **NO PUEDE SUPONERSE** que los directorios *home* de los usuarios están en /home (ni que, en el caso de Solaris, estén en /export/home). Se obtendrán del fichero del sistema adecuado
- ▶ El **MISMO script** tiene que correr **SIN MODIFICACIONES** en linux, BSD y solaris

## Práctica 6: Automatización

- ▶ En ningún caso el script puede dar error en tiempo de ejecución
- ▶ Cada copia de seguridad de un usuario consistirá en un archivo `tar.ext` (`ext` depende del compresor utilizado) conteniendo el directorio *HOME* de dicho usuario (con sus contenidos).
  - ▶ El nombre de dicho archivo será *\*usuario.tar.em* ext, donde *usuario* es el nombre (login) del usuario. Por ejemplo, para el usuario *trump* la copia de seguridad será el archivo *\*trump.tar.gz*. (\* puede ser cualquier grupo de caracteres ascii)
  - ▶ Pueden crearse directorios en `$BACKUP_DIR`
  - ▶ se comprobará que hay espacio suficiente ANTES de realizar la copia de seguridad de cada usuario.

## Práctica 6: Automatización

- ▶ Las copias serán accesibles UNICAMENTE por dicho usuario (y el root naturalmente)
- ▶ El *script* no realizará más de una copia de seguridad al día para cada usuario, aunque se ejecute varias veces en el día
  - ▶ es decir, si fuese colocado en el crontab para ser ejecutado cada dos horas, cada día solo haría una copia de seguridad, en lugar de 12
- ▶ Cada copia de seguridad de un usuario realizada se notificará al log (user.notice), mediante el programa *logger*
- ▶ Cuando no pueda realizarse una copia se notificará al log (user.warning), junto con el motivo mediante el programa *logger*

## Práctica 6: Ayudas

- ▶ El comando `seq` genera una secuencia de números
- ▶ En linux el programa `mkpasswd` suele estar en el paquete *whois* o en el paquete *mkpasswd*. Es conveniente instalarlo
- ▶ Se suministra el script `pass.expect` para *expect* que podría ser útil en Solaris 11

```
#!/usr/bin/expect -f
if {$argc!=2} {
    send_user "uso: $argv0 usuario password \n"
    exit
}
set user [lindex $argv 0]
set password [lindex $argv 1]
spawn passwd $user
expect "New Password: "
send "$password\r"
expect "Re-enter new Password: "
send "$password\r"
send "\r"
expect eof
```