

La última versión de este documento se puede descargar de <https://alvarogonzalezsotillo.github.io/apuntes-clase/sistemas-gestores-bbdd-asir2/apuntes/3/asgbd-03-trabajo-scripts.pdf>

## 1. *Scripts* de inicio y parada de Oracle (1 punto)

Crea dos *scripts* para iniciar y parar **Oracle**

- /home/alumno/scripts/oraclestart.sh
- /home/alumno/scripts/oraclestop.sh

## 2. Arrancar automáticamente Oracle cuando se inicie el servidor (2 puntos)

- **Oracle** debe levantarse cuando la máquina se inicie, y apagarse cuando la máquina se cierre.
- Oracle se iniciará solo si se indica en el fichero /etc/oratab.
- En el fichero /home/alumno/logs/oracle.log se dejará una traza de cuando se arrancó y se paró la máquina, y si fue necesario arrancar o parar **Oracle**. Por ejemplo:

```
2017-02-10-12:40:00 - Solicitud de arrancar Oracle
2017-02-10-12:40:01 - Oracle arrancando porque /etc/oratab indica Y
2017-02-10-12:40:20 - Oracle arrancado
```

Listado 1: Ejemplo de /home/alumno/logs/oracle.log cuando **Oracle** se arranca

```
2017-02-10-12:41:00 - Solicitud de parar Oracle
2017-02-10-12:41:20 - Oracle parado
```

Listado 2: Ejemplo de /home/alumno/logs/oracle.log cuando **Oracle** se para

- Cuando no se arranque **Oracle**, porque se indica N en oratab, **systemd** intentará limpiar el servicio invocando la parada, por lo que puede que se invoque `systemctl stop` automáticamente y aparezca en el *log*

```
2017-02-10-12:40:00 - Solicitud de arrancar Oracle
2017-02-10-12:40:01 - Oracle no se arranca porque /etc/oratab indica N
2017-02-10-12:40:02 - Solicitud de parar Oracle
2017-02-10-12:40:03 - Oracle parado
```

Listado 3: Ejemplo de /home/alumno/logs/oracle.log cuando **Oracle** no se arranca

### Aviso

Los *scripts* no cambian el fichero /etc/oratab, solo lo consultan.

### Aviso

Es posible que **SELinux** no deje arrancar los *scripts*. Para deshabilitarlo temporalmente, se puede usar `setenforce 0`

### 3. Crea usuarios de base de datos (2 puntos)

Crea un script de nombre `/home/alumno/scripts/nuevo-usuario-oracle.sh` que cree un nuevo usuario de oracle. Si se invoca sin parámetros, o con más de dos, mostrará el texto de ayuda del listado 4

```
Crea un usuario nuevo de oracle, con permisos connect y resource.
Si el usuario ya existe, lo desbloquea y le cambia la contraseña.

Uso: nuevo-usuario-oracle.sh <usuario> <contraseña>
```

Listado 4: Ayuda del script nuevo-usuario-oracle.sh

#### Aviso

En la salida del *script* debe quedar claro si el usuario se crea (porque no existe), o solo es desbloqueado.

### 4. Almacena información periódicamente en la base de datos (3 puntos)

Programa un *script* para que cada minuto almacene en la tabla DF la información del comando `df -k`. Esta tabla (listado 5) tendrá como columnas:

- hora: Hora de lanzamiento del comando
- sistema: Nombre del tipo de sistema de ficheros
- tamano: Tamaño en KB del sistema de ficheros
- usado: Tamaño usado, en KB
- montado: Punto de montaje

```
create table DF(
  hora varchar(40),
  sistema varchar(40),
  tamano varchar(40),
  usado varchar(40),
  montado varchar(40)
);
```

Listado 5: Creación de la tabla DF

| Filesystem | 1K-blocks | Used      | Available | Use% | Mounted on        |
|------------|-----------|-----------|-----------|------|-------------------|
| udev       | 4002180   | 0         | 4002180   | 0 %  | /dev              |
| tmpfs      | 804488    | 19756     | 784732    | 3 %  | /run              |
| /dev/sda1  | 237874840 | 183034916 | 42733532  | 82 % | /                 |
| tmpfs      | 4022440   | 437328    | 3585112   | 11 % | /dev/shm          |
| tmpfs      | 5120      | 4         | 5116      | 1 %  | /run/lock         |
| tmpfs      | 4022440   | 0         | 4022440   | 0 %  | /sys/fs/cgroup    |
| /dev/sdb5  | 689521880 | 595546232 | 58926896  | 91 % | /home/windows     |
| cgmsfs     | 100       | 0         | 100       | 0 %  | /run/cgmanager/fs |
| tmpfs      | 804488    | 88        | 804400    | 1 %  | /run/user/1000    |

Listado 6: Ejemplo de salida del comando `df -k`

**Aviso**

Pistas para realizar el *script*:

- Los *heredocs* pueden **contener variables**
- **Cortar columnas** con `awk`
- Leer líneas **una por una** y meterlas en una variable:
- **Quitar la primera línea** de la salida de `df -k` con el comando `tail`
- El *script* debería seguir los siguientes pasos:
  1. Quitar la primera línea de la salida de `df -k`
  2. Leer cada línea con `while` y `read`
  3. Sacar los campos de cada línea con `awk`
  4. Ejecutar una sentencia SQL con los datos extraídos

## 5. Envía un correo periódicamente (2 puntos)

- Programa un *script* para que cada minuto envíe un correo con la información promedio del comando `df -k`. Puedes usar como base para la consulta el listado 7
- El correo se enviará a `alvaro@alvarogonzalez.no-ip.biz`
  - Con copia a `alvaro@debian12-alvaro.local`
- Indica tu nombre en el asunto del correo
- El fichero tendrá **formato CSV**. Se debe poder abrir directamente con **excel** y visualizar su resultado en filas y columnas.

```
1 select
2   sistema, avg(tamano), avg(usado), montado
3 from
4   DF
5 group by
6   sistema, montado;
```

Listado 7: Consulta tipo para extraer información promedio

Para ser un servidor de correo, se necesita **instalar Postfix**

```
1 sudo dnf install -y postfix
2 sudo dnf remove -y sendmail #opcional
3 sudo alternatives --set mta /usr/sbin/sendmail.postfix #opcional
4 sudo systemctl start postfix
```

## 6. Instrucciones de entrega

- El ejercicio se realizará y entregará de manera individual.
- El profesor comprobará el funcionamiento del sistema el día indicado.
- Sube en la tarea del aula virtual un ZIP con todos los ficheros que has creado o modificado:
  - *Scripts*

- *units* de systemd
- Ficheros de cron / anacron