

Administración de Sistemas. Práctica 1

Introducción

Esta primera práctica consiste en la instalación de un sistema operativo y la realización de algunos pequeños scripts para familiarizarse con el intérprete de comandos bash. La práctica se divide en 3 partes: 1) Instalación Sistema Operativo Debian, 2) Primeros pasos con bash y 3) Instalación de aplicaciones en Debian. Las 2 primeras partes deben hacerse en paralelo para completar la práctica en las 2 horas de la sesión.

Al finalizar la práctica deberás entregar un fichero de texto sin formato (extensión .txt) o con markdown (extensión .md) con las respuestas a las preguntas que se van planteando. La entrega de este guión es obligatoria, aunque no contará para la nota final de prácticas. Puedes consultar documentación de markdown en múltiples sitios como <http://commonmark.org/help/>.

A lo largo de las prácticas de esta asignatura vais a utilizar varias máquinas virtuales y es posible que tengáis algún problema con ellas. Por ello os recomendamos la utilización de un repositorio donde tengáis guardado por un lado los ficheros de configuración de las máquinas que vayáis modificando junto con las notas y comentarios que consideréis oportuno y por otro los scripts que iremos haciendo durante las sesiones de prácticas. Una opción es utilizar git con el repositorio institucional de la universidad. En el siguiente enlace hay un buen manual de git, <https://git-scm.com/docs/user-manual> y en este otro enlace está la documentación sobre como añadir vuestra clave ssh a gitlab para acceder a repositorios, <https://docs.gitlab.com/ee/ssh/>

Trabajo previo

El hipervisor a emplear durante las prácticas es VirtualBox que se encuentra instalado en las máquinas CentOS del laboratorio L1.02 y que puedes instalar en tu computador¹. Este documento, https://docs.oracle.com/cd/E26217_01/E26796/html/qs-create-vm.html, describe la instalación de una máquina virtual con VirtualBox y su lectura es necesaria antes de la realización de las prácticas.

Además, es necesario descargar la imagen ISO del sistema operativo y traerla al laboratorio en una memoria USB o disco duro portátil para acelerar el proceso de instalación. La imagen de debian 11.6 para arquitecturas AMD64/x86-64 puede descargarse de:

<https://cdimage.debian.org/debian-cd/current/amd64/iso-dvd/debian-11.6.0-amd64-DVD-1.iso>

Una vez descargada la imagen, deberás comprobar mediante una función de hash la integridad del fichero², es decir, que no haya sido modificado o corrompido en la descarga, para ello tienes que ejecutar el comando:

```
central:~/sha256sum <path_to_image>/debian-11.6.0-amd64-DVD-1.iso
```

¹En el Anexo A del presente documento se ha incluido un resumen sobre los aspectos básicos de las máquinas virtuales

²Este aspecto será tratado en el bloque de Seguridad Básica de la asignatura

Y la salida deberá ser:

```
55f6f49b32d3797621297a9481a6cc3e21b3142f57d8e1279412ff5a267868d8
```

Una vez verificada la corrección de la imagen, ya se puede continuar con la instalación.

PARTE 1: Instalación del sistema operativo GNU/Linux debian

Puedes instalar la máquina virtual tanto en las máquinas del laboratorio como en tu computador personal. En el caso de utilizar vuestro computador personal (portátil u otro), tener en consideración los consejos de configuración del Anexo B. En el primer caso como la cuota de las cuentas está limitada es conveniente utilizar un dispositivo USB externo.

El primer paso de la instalación es lanzar VirtualBox, luego hay que añadir una nueva máquina virtual, en el nombre, tipo y versión podemos escribir **as_2022_2023**, Linux y Debian (64-bit). Como características de la máquina virtual utilizaremos **1GB de memoria** y **4GB de disco duro tipo VDI y con capacidad dinámica** (dynamically allocated)³.

Completado el paso anterior hay que acceder dentro de los ajustes a almacenamiento para que la máquina virtual pueda ver el disco de instalación. Dentro de este menú, añadiremos la imagen iso al controlador IDE y así al arrancar la máquina virtual lanzará el proceso de instalación.

Lanzado el proceso de instalación, tendréis que elegir el idioma, país, formatos varios según la localización (locale), nombre de la máquina (hostname), y poner password de root, y nombre y password de un usuario.

El siguiente paso es el particionado del disco donde deberemos dejar todos los ficheros en la misma partición, y **crear una partición de swap de 512 MB y el resto para la partición raíz que será arrancable y del tipo ext4**. Después tendremos que confirmar los cambios en el disco.⁴

Cuando os pregunte si queréis utilizar una réplica de red, contestad que si. Esto permite la instalación de software desde repositorios de Internet⁵.

En la selección de paquetes **no hay que instalar el escritorio gráfico** ya que el proceso de instalación se alarga bastante más, con las opciones básicas propuestas es suficiente (servidor ssh y software básico). Posteriormente pueden ser instalados escritorios como xfce4, gnome... con apt install xfce4 por ejemplo, aunque en esta asignatura no los vamos a necesitar.

PARTE 2: Primeros pasos con bash

³Como se ha explicado en el anexo A, la máquina virtual tendrá su propio hardware virtualizado. Los archivos con extensión vdi son utilizados por Oracle VM VirtualBox para almacenar el contenido que las máquinas virtuales percibirán como sus discos duros.

⁴En el bloque de almacenamiento de la asignatura abordaremos las cuestiones relacionadas con los sistemas de ficheros y particionado de discos

⁵Ver el Anexo C sobre la instalación de aplicaciones

Mientras se instala Debian, puedes abrir una terminal linux en el computador host, laboratorio o propio, para empezar a realizar pequeños scripts con bash. Si ya has instalado la máquina virtual puedes realizar esta parte en la máquina recién instalada.

Cuando se instala una máquina UNIX/Linux suele contar únicamente con los editores *vi* y, eventualmente, *nano*. Aunque el editor más universal (estándar Posix) en cualquier Unix (Linux, OSX Macintosh, FreeBSD, OpenBSD, etc), sobre todo para un administrador es *vi*. La característica principal de *vi* es que tiene varios modos de funcionamiento. Los 2 principales son: modo *normal* o *comando*, en donde podemos ejecutar comandos tanto del editor como del sistema; y el modo *inserción*, en el que se edita propiamente el texto. La tecla “*escape*” activa el modo normal y la tecla “*i*” activa el de inserción. Otro aspecto que interesa de *vi* es que sus “comandos son utilizados por otras herramientas de Unix, que os servirán para programación shell (como la herramienta *sed*). En esta guía encontrareis una explicación más detallada sobre *vi*, <https://www.ibm.com/developerworks/ssa/linux/library/l-1pic1-v3-103-8/index.html>, o para una versión más sofisticada de *vi*, como es vim, en <http://es.tldp.org/Tutoriales/doc-tutorial-vim/Guia-Vim.pdf>. Adicionalmente, *tenéis disponible en moodle una chuleta* con los principales comandos de *vi*.

El primer paso a realizar es un pequeño script que escriba por pantalla: “hola mundo!”. Para ello utilizaremos uno de los editores y crearemos el fichero `primer_script.sh`. Por ejemplo con vim:

```
as@debian:~$ mkdir practica1 # creamos un directorio para la práctica
as@debian:~$ cd practica 1 # accedemos al directorio creado
as@debian:~$ vim primer_script.sh
```

El contenido del fichero `primer_script.sh` deberá ser:

```
#!/bin/bash
echo “hola mundo!”
```

La primera línea incluye el shebang, par de caracteres `#!/`, y el intérprete de comandos, shell, con el que el script debe ser ejecutado.

Pregunta: Busca la llamada al sistema que se emplea para ejecutar un script y explica su funcionamiento

La segunda línea incluye el comando `echo` que se utiliza para escribir por pantalla. De hecho, `echo` escribe todos sus argumentos por pantalla. Una vez escrito el script puedes guardarlo y salir en vim con `escape` para cambiar de modo, “`:`” para pasar a escribir un comando y `wq` para escribir y salir del editor.

Luego puedes probar a ejecutar el script con `./primer_script.sh` y ocurrirá un error.

Pregunta: ¿A qué se debe el error? ¿Cómo podría arreglarse el problema?

Para solucionarlo tendrás que añadir permisos de ejecución al script con `chmod u+x primer_script.sh` o pedirle a `bash` que lo ejecute directamente con `/bin/bash primer_script.sh`. Debes probar ambas opciones.

Además de tener comandos propios como `echo`, `bash` dispone de otros comandos para generar estructuras de control con bucles. Para probarlos vamos a escribir un segundo script, `segundo_script.sh` cuyo contenido será:

```
#!/bin/bash
for i in hola mundo
do
    echo "${i}"
done
```

Dejando fuera el generador de bucles que itera sobre las cadenas hola y mundo, este script tiene una variable de nombre i. En bash, cuando queremos acceder una variable siempre le añadiremos el prefijo \$ y opcionalmente podremos utilizar las llaves. Para asignar un valor a una variable no es necesario el prefijo.

Pregunta: ¿Cuál es la salida del script segundo_script.sh? ¿Por qué aparecen 2 líneas en vez de una?

Pregunta: Añade una nueva variable final_cadena que contenga la cadena: “ , como estas?” e imprímela dentro del mismo comando echo.

Una de las capacidades más útiles de bash, y de los shells en general, es la concatenación de comandos. Con bash resulta muy fácil que la salida de un comando sea la entrada de otro o “capturar” la salida dentro de un script.

En el siguiente ejemplo vamos a capturar la salida del comando ls y mostrar el contenido de los ficheros.

```
#!/bin/bash
for i in $(ls)
do
    cat "${i}"
done
```

En este caso, la iteración se producirá sobre la salida del comando ls mediante un subshell, instanciado mediante \$(...) y para cada fichero, se mostrará su contenido por pantalla mediante el comando cat.

Pregunta: ¿Se produce algún error si añadimos un subdirectorio en el directorio actual?

Otra característica muy útil del Shell es la sustitución de nombres de ficheros mediante caracteres comodín.

Pregunta: Reescribe el último script utilizando caracteres comodín, en lugar de utilizar el comando ls.

PARTE 3: Instalación de aplicaciones

Una vez completados las partes 1 y 2, se va a proceder a la instalación de un paquete⁶. Dicha instalación requiere de privilegios de administración de los que el usuario creado (as en los siguientes ejemplos) no dispone por defecto. Por ello primero instalaremos la aplicación sudo (substitute user do). Sudo permite ejecutar programas con privilegios de seguridad de otro usuario, como por ejemplo root. Al ejecutar sudo, los usuarios deben confirmar su identidad dando su propia contraseña antes de la ejecución del programa requerido. En otras palabras, una vez se ha autenticado el usuario, y si el archivo de

⁶Ver el Anexo C sobre la instalación de aplicaciones

configuración `/etc/sudoers` permite dar al usuario acceso al comando requerido, el sistema lo ejecuta y lo registra en `/var/log/auth.log`.

Para ver si `sudo` está instalado escribiremos el siguiente comando:

```
as@debian11:~$ apt list --installed sudo
```

En caso negativo, deberemos instalarla y para ello primero utilizaremos el comando `su` para cambiar al usuario `root` y luego utilizaremos `apt` para la instalación.

```
as@debian:~$ su root
root@debian:~$ apt update
root@debian:~$ apt install sudo
```

Una vez instalado, el primer paso será comprobar en el fichero `/etc/sudoers` a qué grupo tiene que pertenecer un usuario para poder ejecutar comandos como administrador, `root`. Conocido este grupo deberá ejecutarse el comando `adduser`, es recomendable hacer `man adduser` para conocer su funcionamiento, para añadir a nuestro usuario a dicho grupo.

```
root@debian:~$ /sbin/adduser <tu_nombre_de_usuario> sudo
```

Es necesario **salir de la sesión y volver a entrar** para que los cambios tengan efecto una vez añadido el usuario al grupo.

Después de salir y volver a entrar al sistema, vamos a comprobar si nuestro usuario realmente tiene privilegios de super-usuario con `sudo`. Para ello, realizaremos una tarea básica de monitorización del sistema que consiste en visualizar un fichero para el cual solo tiene permiso de lectura el administrador.

```
as@debian:~$ cat /etc/shadow
```

Debería aparecer un error, sin embargo, al ejecutar:

```
as@debian:~$ sudo cat /etc/shadow
```

la ejecución debería ser correcta.

Notas finales:

Si encuentras alguna errata en este documento y/o tienes alguna sugerencia de mejora comunícaselo por favor a cualquiera de los profesores de la asignatura.

Anexo A: Fundamentos de las máquinas virtuales

Una Máquina Real es una máquina que contiene componentes hardware que ofrecen una operatividad a muy bajo nivel (nivel físico) en una Arquitectura conocida y donde se pueden obtener los mejores y más eficaces resultados en cuanto a velocidad final de la máquina. Sobre ella se ejecutan tanto Sistemas Operativos como otros Procesos que sólo necesitan de una arquitectura conocida.

Una máquina virtual es un contenedor de software perfectamente aislado que puede ejecutar sus propios sistemas operativos y aplicaciones como si fuera un ordenador físico. Una máquina virtual se comporta exactamente igual que lo hace un ordenador físico y contiene sus propias CPU, RAM, disco duro y tarjetas de distintos interfaces como puede verse en la Figura 1.

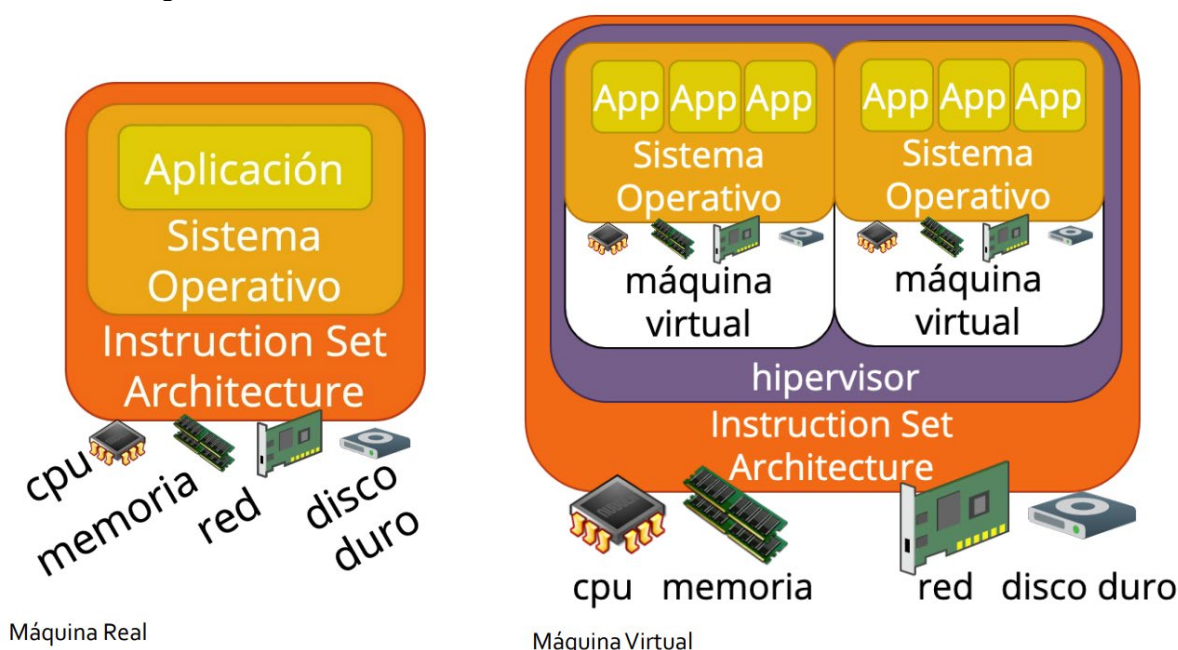


Figura 1. Máquina Real vs. Virtual [Autor: Darío Suárez]

Así, una característica esencial de las máquinas virtuales es que los procesos que ejecutan están limitados por los recursos y abstracciones proporcionados por ellas. Estos procesos no pueden escapar de esta "computadora virtual".

El uso que vamos a darle en la asignatura es administrar sistemas operativos. De esta forma podemos modificar la configuración de un sistema operativo sin miedo a que se desconfigure el sistema operativo primario.

El hipervisor o Virtual Machine Monitor (ver Figura 1) es la capa software que hace de intermediario entre las máquinas virtuales que aloja (sus sistemas operativos) y el hardware real. Además, presenta un hardware virtual al sistema operativo virtualizado.

Existen dos tipos de hipervisores, los que se ejecutan directamente sobre el hardware y los que se ejecutan sobre un sistema operativo como es el caso de VirtualBox.

Anexo B: Configuración de vuestro ordenador personal para ejecución de máquinas virtuales

En primer lugar, si tenéis un sistema Windows, se aconseja, vehementemente, la instalación de una distribución (una variante de sistema) de Linux en vuestro propio ordenador. Sea como sistema único o como instalación dual (conviven Linux y Windows en el mismo ordenador). Las distribuciones de Linux que se aconsejan instalar son, principalmente, Xubuntu y, eventualmente, Manjaro. El primero es un derivado de Debian y el segundo un derivado de Archlinux. Cualquiera de ellos, tiene posibilidad directa y fácil de instalar el paquete software de Virtualbox que tiene disponible en su propio gestor de paquetes software. El disponer de una distribución de Linux en vuestro ordenador a primer nivel, no sólo es interesante para esta asignatura, sino también para otras asignaturas que cursareis en los estudios del Grado en Informática.

Además, en cualquier caso, deberéis configurar la BIOS de vuestro ordenador con el soporte de **virtualización activado** y el soporte “**Secure BOOT**” **desactivado**.

Anexo C: Instalación y gestión de software

A la hora de instalar aplicaciones, éstas lo hacen a través de un paquete. Un paquete es un fichero comprimido (.ar) que contiene los archivos de una aplicación (o biblioteca, cabeceras, etc.).

Un paquete incluye además meta-datos que contienen información como una descripción, versión, requisitos previos, scripts de instalación, etc.

También se ocupa de gestionar las dependencias de ese paquete como las bibliotecas que la aplicación utiliza u otras como ficheros de configuración, drivers de dispositivos, bases de datos, etc. Algunos posibles niveles de dependencias son:

- *depends*: El paquete X requiere del paquete Y para funcionar
- *recommends*: El paquete X funcionará mejor con el paquete Y
- *suggests*: El paquete Y suele mejorar la funcionalidad del paquete X
- *conflicts*: El paquete X no funciona con el paquete Y instalado

Los paquetes tienen multitud de formatos dependiendo del sistema operativo, para Debian y derivados como Ubuntu o Mint son archivos .deb (Debian Software Package,) binarios y .dsc (Debian source control) fuentes. También existen paquetes virtuales que no instalan archivos pero permiten, por ejemplo, agrupar muchos paquetes (útil cuando instalamos aplicaciones que requieren muchos paquetes como un escritorio) o enlazar a la última versión.

Existe un convenio para nombrar los paquetes:

`<Name>_<Version>.<Revision>_<Architecture>.deb`

El formato del paquete es ar, muy similar a tar. Desempaquetar con `ar x archivo.deb` extrayendo:

- `debian-binary`: fichero de texto con la versión del formato .deb (2.0)
- `control.tar.gz`: metadatos.
- `data.tar.gz`: los archivos ejecutables, datos, librerías de la aplicación.

`tar xvzf control.tar.gz`:

- `./postinst`: scripts shell de post-instalación
- `./control`: información y descripción del paquete, versión, dependencias...
- `./conffiles`: lista de archivos de configuración
- `./prerm`: script de shell de pre-eliminación
- `./postrm`: script shell de post-eliminación
- `./preinst`: script shell de pre-instalación
- `./md5sums`: checksums MD5

Además, estos paquetes pueden ir firmados digitalmente para garantizar su autoría.

El comando `dpkg` en Linux es una herramienta utilizada para administrar paquetes a través de su instalación, eliminación y compilación. En una distribución basada en Debian, Linux preconfigura el gestor de paquetes `apt` o `apt-get` por paquete, el cual a su vez, destina al comando `dpkg` para que se encargue de la gestión de paquetes de Debian.

Este comando aunque no resuelve dependencias, permite la opción de obtener información sobre los paquetes. Además, cumple funciones como la instalación, configuración y eliminación de archivos instalados en el sistema.

Los paquetes se encuentran almacenados en repositorios que pueden estar online, en un CD/DVD, etc. Aquí es donde se pueden buscar e instalar paquetes ofrecidos por los desarrolladores de la distribución o de la aplicación.

Las distribuciones de Linux vienen con una lista de repositorios preconfigurados oficiales, pero el administrador del sistema puede añadir otros. Casi todas las distribuciones instalan aplicaciones, bibliotecas, etc., desde repositorios. Existen distintos tipos de repositorios como los oficiales, mirrors (que son copias de los oficiales para no sobrecargarlos), o de aficionados.

Las herramientas de gestión de paquetes suelen descargar de los repositorios facilitando la instalación de paquetes, descargando e instalando de manera automática las dependencias. Y también la actualización automática de paquetes a las últimas versiones disponibles.

La herramienta apt-get (Advanced Packing Tool) se configura en /etc/apt/apt.conf y /etc/apt/apt.conf.d. En /etc/apt/sources.list y /etc/apt/sources.list.d se encuentra la url de los repositorios. En /var/cache/apt y /var/lib/apt contienen paquetes para que las búsquedas se realicen en los archivos locales.

Uso de apt-get:

- apt-get install *paquete1 paquete2* ...
- apt-get remove *paquete1 paquete2*...
 - apt-get purge *paquete*: elimina también la configuración

Apt-get autoremove (quita paquetes no usados)

Apt-get update

- refresca paquetes disponibles en repositorios
- necesario si añaden/quitan repositorios en sources.list.d

Apt-get upgrade

- actualiza todos los paquetes instalados

Apt-get clean

- limpia la caché de paquetes descargados
- apt-get autoclean: sólo de paquetes no instalados

Apt-cache search *termino* (busca en nombres y descripciones de paquetes disponibles)

Apt-cache show *paquete* (muestra información del paquete)

Apt-cache policy *paquete* (muestra versión instalada, y versiones disponibles)

Add-apt-repository ppa:<repository-name> (añade repo desde Launchpad personal Package Archive que son paquetes de terceros)

Apt-key add *archivo_clave* (gestión de claves para firma digital)

También se podrían instalar los paquetes desde las fuentes. Son ficheros normalmente .tar.gz o tar.bz2. Los pasos típicos son emplear ./configure o cmake para generar el make file, luego compilamos y después se instala. Las dependencias se gestionan a mano siendo una de las partes más complicadas. Aunque existen herramientas como auto-apt run ./configure que va a intentar buscar las dependencias directamente.

En general, para la instalación de paquetes suele ser necesario permisos de root (sudo), para aplicaciones disponibles para todos los usuarios.

¿Donde se instala el software en un sistema Linux?

- /bin, /lib, /sbin, etc. reservado para aplicaciones y paquetes esenciales para el sistema
- /etc para archivos de configuración globales

- /usr (/usr/bin, /usr/lib, /usr/shar, ...) para aplicaciones, bibliotecas, documentación, etc. instaladas desde paquetes, incluidos con la distribución, del gestor de paquetes, y muchas veces al compilar desde fuentes (desde fuentes es posible cambiar el prefijo de instalación).
- Usrlocal (/usr/local/bin, /usr/local/lib,...) para software ajeno a la distribución, instalado manualmente, o desde fuentes.
- /opt/*nombrepaquete*: externos a la distribución, y que tienen todos sus archivos bajo un mismo árbol de directorios.

Hay herramientas gráficas como synaptic que son un GUI para (apt-get, apt-cache, ...).

Fecha de última modificación: 6 de February de 2023