

5€ DE BIENVENIDA

Comandos Fundamentos de Computadores

Práctica 1: Introducción al Shell de Linux

cd ruta_directorio: cambio del directorio actual al indicado por la ruta dada

cd ó cd ~ : cambio al directorio de inicio usuario (/home/alumno/)

cd - : cambio al último directorio al que habíamos estado

ls: lista el contenido del directorio actual

pwd: muestra donde estamos

.. : retroceder en la ruta

whoami: nombre de usuario

touch FICHERO: crea nuevo fichero vacío en la ruta indicada

cat FICHERO ... : muestra el contenido de los ficheros indicados

cp: copia un archivo

rm FICHERO ... : borra los ficheros indicados

mkdir DIRECTORIO: crea directorio

rmdir DIRECTORIO: borra el directorio indicado si está vacío, si no, usar **rm -r**

exit: terminar la sesión actual

date: fecha y hora actual

clear: limpia el terminal

reset: resetea el terminal

script: genera un fichero que graba el terminal llamado typescript y borra cualquiera que se llame igual o no tenga nombre

script -a NOMBRE: continúa o genera un fichero con nombre

exit: finaliza la grabación

less -r typescript: muestra el contenido del fichero

Ruta absoluta: empiezan con / ó ~

sudo apt [...]: necesitamos poner sudo para que lo que vayamos a instalar o actualizar lo haga el superusuario

apt update: actualiza la base de datos local de paquetes disponibles

apt list upgradable: muestra una lista de los paquetes actualizables

apt upgrade: actualiza los paquetes actualizables

apt list nombre-paquete: busca el paquete con nombre *nombre-paquete* en los repositorios disponibles o entre los paquetes instalados

apt install nombre-paquete: instala el paquete *nombre-paquete* desde los repositorios disponibles

Práctica 2: Control de versiones con Git

git clone url: clona un repositorio remoto con la dirección *url* en el directorio actual

git pull: obtener últimos cambios que han sido publicados en el repositorio

git add NOMBRE_FICHERO: inserta el fichero en el índice del repositorio

git commit: actualiza cambios en el repositorio local

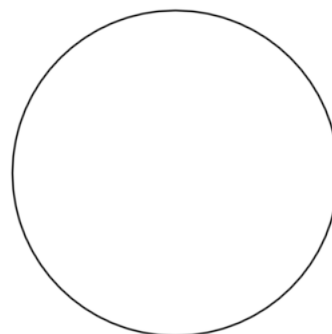
git push: actualiza el repositorio remoto según el estado del repositorio local

git status: comprueba el estado del repositorio

git diff: observar los cambios que se han hecho

git clean -n: elimina un archivo de nuestro directorio de trabajo

git push origin main: actualizar el repositorio remoto



Práctica 3: Representación de la información

FFFFFF = BLANCO → Cuando todos los colores tienen el valor máximo, el píxel resultante es blanco.

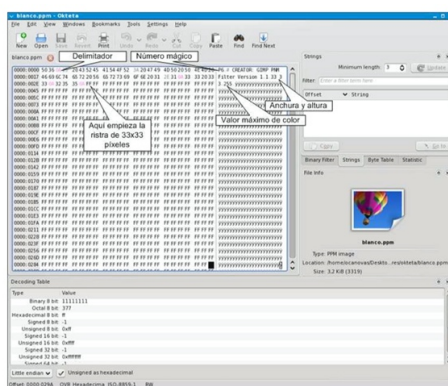


Figura I.5: Fichero PPM en Okteta.

Fórmula para calcular los píxeles:

Práctica 4: Sistemas de ficheros en el Shell de Linux

. : fichero oculto

Ctrl + mayus + C / V: copiar / pegar

whoami: muestra nombre de usuario

who: muestra los usuarios actualmente conectados al sistema

su: cambia de usuario normal a superusuario (*root*)

sudo orden: ejecuta el comando *orden* como superusuario

man orden: manual de uso de la orden tecleada y se sale de ella usando **q**. Para buscar una palabra clave usamos /seguido de la cadena a buscar

exit: salir

groups: podemos ver a qué grupos pertenecemos

ls: lista

- **ls -l** : listado largo y detallado
- **ls -a** : listado incluyendo ficheros ocultos
- **ls -R** : lista recursivamente mostrando el contenido de todos los subdirectorios, a cualquier profundidad
- **ls -S** : lista en orden de tamaño
- **ls -t** : lista en orden de fecha y hora de la última modificación

*****: cero o más caracteres cualesquiera

?: carácter cualquiera

[a – z]: cualquier carácter del rango indicado

[!b – d]: cualquier carácter que no esté en el rango indicado

{nom1, nom2, ...}: cualquier secuencia de caracteres o nombres de la lista

cat FICHERO: muestra el contenido de golpe en el terminal

less FICHERO: muestra el contenido con posibilidad de *scroll* y búsqueda de palabras con */palabra*

hexdump -C ARCHIVO: vuelca el archivo en formato hexadecimal en el terminal

touch nombre_fichero: crea un nuevo fichero llamado *nombre_fichero* vacío

cp fichero_origen_1 [fichero_origen_2 ...] fichero_destino|dir_destino: copia un fichero a otra ubicación.

También puede copiar directorios, incluyendo su contenido.

mv fichero_origen1 fichero_destino: similar a cp pero borra el contenido del origen y lo deja solo en el destino.

cp -r direc_orig direc_dest: copia recursivamente todo el contenido del directorio origen al destino

mkdir -p dir1/dir2/...: crea el directorio 1 y dentro de él el directorio 2 etc
rm -f: sirve para borrar sin confirmación
echo "string": sirve para añadir texto o mostrar texto en el terminal

Gestión de los permisos

chmod 3cifrasOctal fichero|directorio: cambia los permisos. (754 = 111 101 100 = rwx r-x r--)
chown nuevo_propietario fich|dir: cambia el propietario de un fichero. Se necesita ser superusuario para que este comando se pueda ejecutar con éxito. También permite cambiar el grupo propietario del fich o dir.
chgrp nuevoGrupo fich|dir: cambia el grupo de un fichero o directorio. Se necesita ser superusuario.

Búsqueda de ficheros

find [ruta1 ruta2 ...] expresión: busca recursivamente entradas que concuerdan con lo que buscamos.

- -iname nombre_fichero_a_buscar
- -type f/d
- -user usuario
- -mtime [+/-]n. Fichero modificado de +/- n días
- -size [+/-] n. Fichero con un tamaño de .. 512xn bytes (-size +28k)
- -printf "...". Sirve para imprimir cadenas de texto. Usamos %p = ruta, %s = tamaño, %u = usuario propietario. (find . -type f -printf "El fichero %p ocupa %s bytes\n")

Podemos usar operadores lógicos para combinar los criterios de búsqueda:

- ! negación
- -o disyunción
- -a conjunción (AND)

find /etc/default fc-alumno -type f -name "dir_*" → búsqueda en /etc/... y en fc-alumno los ficheros que empiezan por dir_

Búsqueda de texto en ficheros

grep [opciones] "patrón" [ruta ...]: localizar cierta palabra o frase en un fichero

- [patrón] -- ^ (inicio de línea), \$ (fin de línea)
- -i : busca ignorando el uso de mayúsculas
- -v : invierte la búsqueda seleccionando las líneas que NO concuerdan con el patrón dado
- -l : muestra el nombre del fichero que contiene la línea coincidente
- -n : muestra el número de línea para cada línea coincidente, empieza en 1
- -w : busca únicamente palabras completa, no fragmentos
- -r : búsqueda recursivamente en los directorios

grep -r "^[A-Z].s" fich.txt → busca recursivamente las líneas del fichero indicado que empiezan con una letra mayúscula, seguida de cualquier carácter y a continuación la letra 's'

Compresión y descompresión de ficheros

tar czvf result.tar.gz direc|fich: empaqueta y comprime los directorios y ficheros indicados como segundo y sucesivos parámetros

tar tzvf result.tar.gz: lista en el terminal los contenidos del fichero result.tar.gz

tar xzvf result.tar.gz: descomprime y desempaqueta el fichero result.tar.gz recuperando los ficheros, permisos y la estructura de directorios original.

tar xzvf result.tar.gz -C ./otrodir: si deseando extraer los ficheros en un directorio diferente al de trabajo actual se debe usar la opción -C

Enlaces

Físicos o duros

ln fichero_a_enlazar enlace: los enlaces duros se caracterizan porque hay que borrar los enlaces y también el fichero enlazado para que desaparezca completamente el fichero

Simbólicos o blandos

ln -s fichero_a_enlazar enlace: los enlaces simbólico se caracterizan porque si borramos el fichero enlazado original, los enlaces quedan "colgando", y el contenido del fichero se pierde

5€ DE BIENVENIDA

Espacio en disco

df : informa de espacio, total, libre y ocupado en todos los sistemas de ficheros montados

du -hs directorio1 ...: muestra los bloques que ocupan realmente en disco el directorio indicado y todo lo que cuelga de él

Práctica 5: Procesos en el Shell de Linux. Tuberías y filtros

Lanzamiento en primer plano

comando1 ; comando2 ; ...: lanza procesos en secuencia pero siempre en primer plano. Hasta que no acaba el último no podemos recuperar el prompt del Shell.

Con **Ctrl-C** podemos matar y con **Ctrl-Z** se duerme en el proceso en primer plano.

fg : despierta el programa y lo devuelve a primer plano de nuevo

bg : despierta el programa y lo pasa a segundo plano

Lanzamiento en segundo plano

comando1 & comando2 & ...: lanza procesos en secuencia pero siempre en segundo plano. El Shell imprime el PID de los procesos

Monitorización y control de procesos

ps [-Af]: el comando **ps** muestra un listado de procesos activos

kill [-9] PID

pgrep PATRÓN: busca entre los procesos en ejecución y nos devuelve su PID

kill PROGRAMA: termina el programa. Si no indicamos nada se ejecutará la SIGTERM 15 pero, podemos indicar otra con **kill -signal num_signal PROGRAMA**

top : proporciona una vista dinámica en tiempo real de un sistema en ejecución, mostrando información de resumen del sistema. Si estamos haciendo un script, debemos pararlo ya que se nos borrará el terminal como si hiciésemos un **clear**.

htop : alternativa a **top** con una interfaz más amigable que permite usar el ratón para seleccionar y desplazarse

glances : muestra una gran cantidad de información

Entrada y salidas de un proceso. Redireccionamiento

< stdin (entrada estándar), **> stdout** (salida estándar), **2> stderr** (salida de error).

ls -l > fich.txt: la salida se escribe en un fichero

ls -l >> fich.txt: en vez de machacar el fichero, añade el contenido al final para no borrar nada

sort < fichero: ordena las líneas por orden alfabético (o numérico, con **-n**)

Podemos redirigir la salida de estándar a la de error y viceversa mediante las combinaciones de caracteres **2>&1** (**stderr a stdout**) y **1>&2** (**stdout a stderr**).

Por ejemplo, cuando queremos realizar una búsqueda con **find** en directorios sobre los que no tenemos permisos de acceso, podemos descartar ese (largo) listado de errores haciendo un redireccionamiento a la salida de error al dispositivo nulo **/dev/null**.

find / -name fichero_que_busco 2> /dev/null

Tuberías y filtros

Usando **|** podemos crear una tubería de manera que por ejemplo:

sort datos > datos_ordenados ; less datos_ordenados → **sort datos | less**

Filtros

Los filtros se pueden combinar mediante tuberías para llevar a cabo una tarea de forma mucho más rápida.

- **wc** : cuenta el número de líneas (-l), palabras, bytes, etc
- **nl** : añade el número de línea al inicio
- **head** y **tail** : extrae las N primeras/últimas líneas (con la opción **--lines=N**)
- **tr** : sustituye, agrupa o elimina caracteres
- **cut** : muestra partes seleccionadas de cada línea
- **rev** : invierte el orden de los caracteres de cada línea

wc -l *.py | sort -n | head -n 1 → se muestra el número de líneas de cada fichero del directorio actual cuyo nombre acaba en **.py** y ordenamos el resultado de la cuenta de líneas de manera que el de menos líneas aparezca primero y por último tomamos únicamente la primera línea producida como salida de la ordenación.

Con esta promo,
te llevas **5€** por
tu cara bonita al
subir **3 apuntes**

a Wuolah
Wuolah



Subshells

Esta es otra técnica para combinar comandos en una misma orden. Lo veremos con el siguiente ejemplo:

wc -l \$(find . -type f -name "*.py") : el shell ejecuta primero lo que hay entre paréntesis, es decir, el find y luego se le pasará a wc.

Se suelen utilizar los subshells para combinar el grep y el find:

grep -w "main" \$(find .. -name "*.c") : se busca la palabra main en todos los ficheros cuyo nombre acaba en .c situados dentro directorio padre al actual, a cualquier profundidad.

Variables de entorno. PATH

env : lista de variables de entorno

Se puede mostrar el valor de cualquier variable de entorno con **echo \$*variableEntorno*** y se puede modificar el valor de esta mediante ***variableEntorno*=*nuevoValor***

Práctica 6: Monitorización del sistema en Linux

El sistema de ficheros virtual /proc

/proc/cpuinfo : muestra información acerca del procesador

/proc/meminfo : muestra información sobre la memoria física

/proc/<PID>/: directorio de un proceso en ejecución donde encontramos información gracias a:

- **cmdline** : contiene el comando emitido al iniciar el proceso
- **cwd** : enlace simbólico al directorio de trabajo actual del proceso
- **exe** : enlace simbólico al fichero ejecutable del programa ejecutado por este proceso
- **fd** : directorio q contiene todos los descriptores de fichero para este proceso. Los descriptores se muestran mediante enlaces numerados
- **maps** : lista que muestra el uso del espacio de direcciones virtual del proceso
- **status** : estado del proceso, incluyendo uso de memoria

Para poder visualizar los archivos **/proc/...** debemos usar el comando **less**.

Monitorizando el espacio de direcciones virtual de un proceso: /proc/<PID>/maps

/proc/<PID>/maps ó **pmap** : muestra el mapa memoria

Monitorizando las llamadas al sistema efectuadas por un proceso

strace : ilustra las interacciones entre los procesos y el kernel de Linux.

Con la opción **-o** podemos imprimir las cosas en un fichero específico.

Con la opción **-trace=...** es posible indicar qué llamadas al sistema concreta se desea tracear

Monitorizando los ficheros abiertos por un proceso: /proc/<PID>/fd

Monitorizando las interrupciones hardware mediante /proc/interrupts

watch -d cat /proc/interrupts : observar el número de interrupciones recibidas a lo largo del tiempo. Podemos usar **grep** para filtrar el contenido. **grep [..] /proc/interrupts**

Práctica 7: Sistema de compilación en Linux. Dependencias

Comprobar la instalación de un programa del sistema

whereis programa

sudo dpkg -L programa : ubicación de los ficheros de un programa instalado por el sistema

Instalación de software a partir del código fuente en Linux

1. **./configure** : esto lo ejecutamos en el directorio raíz de la distribución de código y se generará **Makefile**
2. **make** : ejecutará **Makefile**
3. **make install** : se copiará el programa compilado, sus bibliotecas y su documentación

Mirar ejemplo sobre la instalación de un programa.

Ejemplo

git clone url_programa

ls para buscar un documento que ponga README

sudo apt install ... // apt list ... (para ver si está instalado)

./autogen.sh

./configure --prefix=\$HOME/local/...

make

make install

Python en modo consola / programa

python3 : se imprime las salidas por la consola.

echo "print("Hola")" > hello.py

python3 hello.py

Gestión de paquetes en Python

pip list : listado de paquetes instalados

Entornos virtuales en Python

python3 -m venv [miruta] : crea un entorno virtual en un directorio determinado por la ruta especificada

source [miruta]/bin/activate : antes de poder utilizar un entorno virtual debemos activarlo explícitamente. Esto hace que el entorno virtual sea el intérprete de Python local por defecto mientras dure la sesión por consola

deactivate : para terminar la sesión con el entorno virtual

Jupyter Notebooks con entornos virtuales en local

Tras haber activado nuestro entorno virtual con **source ~/[miruta]/bin/activate**

jupyter notebook

Práctica 8: Lenguajes del computador: alto nivel, ensamblador y máquina

Aplicación web Compiler Explorer

En el repertorio de instrucciones del ISA x86-64 encontramos:

- **push** , **pop** → guardar / recuperar datos en / de la pila
- **call** , **ret** → llamar / regresar procedimientos

Generación de código ensamblador

gcc fichero.c -o fichero_ejecutable : compilación para obtener un fichero ejecutable pero esto no es paso a paso

gcc fichero.c -S : pasar de código de alto nivel a lenguaje ensamblador

gcc -c fichero.s : pasar de código ensamblador a código objeto

objdump -d fichero.o : desensamblador

gcc fichero.o -o fichero_ejecutable : listo para ejecutar

Bibliotecas dinámicas

El ejecutable generado se puede probar tecleando **./fichero_ejecutable**

ldd fichero_ejecutable : comprobamos las bibliotecas dinámicas con las que enlaza nuestro ejecutable generado

Bibliotecas estáticas

Autocontenido que no depende de bibliotecas dinámicas externas tecleando:

gcc -static fichero.o -o fichero.static

Carga en memoria de un fichero ejecutable

gcc -g -static fichero.c -o fichero.static : para que el ejecutable cuente con la información necesaria para que GDB resulte plenamente útil tenemos que añadir la opción **-g**

gdb fichero.static : lanzamos el depurador

- **break** : ejecución del programa se detiene en ese punto
- **layout split** : nos aparecen dos paneles: el código fuente del programa y su traducción a código ensamblador

Aspectos fundamentales del ISA x86-64

RAX – 64 bits ; EAX – 32 bits

Repertorio básico de instrucciones en x86-64

mnemonic source, destination : el operando a la derecha de la coma es el que se modifica como resultado de la ejecución de la instrucción

Instrucciones aritmético-lógicas: sirven para hacer operaciones aritméticas y/o lógicas con los operandos.

add %rbx, %rax # Suma RBX a RAX, y deja el resultado en RAX

sub \$1234, %rax # Resta 1234 a RAX, y deja el resultado en RAX

WUOLAH

Oh Wuolah wuolita
Tu que eres tan bonita