PRÁCTICA 2

Sistema operativo Unix

Q1. ***A través de las páginas del manual (comando man), determinar la funcionalidad de las siguientes opciones aplicadas a su correspondiente comando:***

Ls (mostrar)

−a/ Muestra los ficheros y directorios, incluyendo los ocultos.

−l/ Muestra en pantalla el nombre y las propiedades del fichero como los permisos, el nombre del propietario, el nombre del usuario, nombre del grupo, tamaño, y fecha de la última modificación, etc.

−m/ Muestra los ficheros de forma horizontal, separándolos mediante comas.

−F/ Atribuye una serie de caracteres gráficos y colores a los archivos, dependiendo del tipo de estos.

Ejecutables * (rojo)

enlace simbólico @

directorio / (azul)

−R/ Muestra la lista de los subdirectorios recursivamente.

−help/ Emerge una ayuda relacionada con el comando actual.

Rm (borrar)

−f/ No pregunta al usuario si quiere confirmar la eliminación de un fichero.

−i/ Nos pide confirmación antes de eliminar un fichero.

−r/ Borra el contenido de un directorio.

−v/ Muestra el nombre de cada fichero antes de eliminarlo.

Cp (copiar)

−a/ No modifica la estructura y atributos del archivo de origen en la nueva copia.

−f/ Elimina los ficheros de destino que tengan el mismo nombre.

−i/ Emerge una pregunta de confirmación para sobrescribir los ficheros del destino con el mismo nombre.

−p/ No modifica el dueño, el grupo, el permiso y la fecha de la última modificación en los ficheros de destino respecto los originales.

−R/ Copia los subdirectorios.

Mv (renombrar)

−f/ Elimina los ficheros del origen una vez nombrados los de destino sin confirmación del usuario.

−i/ Nos pide confirmación antes de sobrescribir los ficheros de destino que posean el mismo nombre que los ficheros del origen.

−v/ Nos muestra en pantalla el nombre original antes de cambiarlo.

Q2. ***El comando cp −a ¿es equivalente al comando cp −dpR? ¿Por qué?***

· Sí, porque las opciones −dpR combinadas hacen lo mismo que −a, así −p mantiene las características del fichero y de los directorios y −R mantiene las propiedades del fichero.

Q3. ***Crear los directorios (/usr/bin, (/usr/include, (/usr/doc. En (/usr/bin copiar todos los ficheros de /usr/bin que empiecen por w. ¿Reconoces algunos de los comandos Unix?***

· mkdir (/usr

· mkdir (/usr/bin

· mkdir (/usr/include

· mkdir (/usr /doc

· cp /usr/bin/w * (/usr/bin

Utilizamos los comandos mkdir para crear directorio y cp para copiar los ficheros.

Q4. ***¿Se han modificado los permisos? ¿Quién es ahora el propietario de esos ficheros ¿Podrías borrar los ficheros copiados?***

· Sí, antes el propietario era el root (superusuario) y ahora somos nosotros (users). Sí podemos borrar los ficheros copiados, pero no los originales.

Q5. ***Copiar todos los archivos de /usr/include a (/usr/include. ¿Se han copiado también los directorios o únicamente los archivos? ¿Cómo deberíamos ejecutar el comando cp para que también se copiaran los directorios?***

cp /usr/include/* (/usr/include

· Así solo se copian los archivos, para que se copien los directorios hacemos:

cp −R /usr/include (/usr/include

Q6. ***Copiar todos los ficheros (sin incluir directorios) de /usr/doc/howto/en a (/usr/doc. ¿Qué tipo de ficheros son?***

cp /usr/doc/howto/en/* (/usr/doc

Son archivos comprimidos por medio de gzip (gz).

Q7. ***Crear el directorio (/usr/etc y copiar en él todos los archivos de /etc/skel. Renombrar los ficheros .Xdefaults, .profile y .bashrc de tal forma que deje de ser ocultos.***

mkdir (/usr/etc

cp /etc/skel/*

mv .Xdefaults Xdefaults

mv .profile profile

mv .bashrc bashrc

· Los renombramos quitándoles el punto de delante, pues los archivos ocultos se distinguen por este punto.

Q8. ***Dentro del directorio (/usr/etc ejecutar el comando cat bashrc. ¿Qué hace exactamente dicho comando ?¿Para qué sirven las opciones b, n y T?¿Cuántas líneas tiene en total el fichero bashrc?¿Cuántas líneas escritas tiene?¿Se trata de un fichero binario o de texto?.Los ficheros Xdefaults y profile, ¿son binarios o de texto?***

· Es un visor que visualiza la estructura y el contenido un fichero.

cat −a/ Número de líneas escritas.

cat −b/ Número de líneas en blanco.

cat −T/ Muestra cuando hay un tabulador, mostrándonos ^I en vez del espacio.

· El archivo bashrc tiene 38 líneas en blanco y 51 escritas. Es un archivo de texto.

· Profile es de texto, mientras que Xdefaults es binario.

Q9. ***Dentro del directorio (/usr/etc ejecutar el comando less bashrc. ¿Qué hace exactamente dicho comando?. Si una vez ejecutado el comando, pulsamos ^B, ¿ qué ocurre?. Determinar la funcionalidad de ^F.***

· El comando less, hace exactamente lo mismo que el comando cat, pero este nos dice cuanto % de fichero nos queda por ver.

^B, hace retroceder en % tanto como quepa en la pantalla.

^F, hace adelantar en % tanto como nos quepa en la pantalla.

Q10. ***En el directorio (/usr/doc ejecutar el comando cat Modem−HOWTO.gz ¿Por qué el comando no se ejecuta correctamente? Ejecutar el comando zless Modem−HOWTO.gz. ¿Funciona? ¿Qué hace exactamente el comando zless?¿Cuál es la funcionalidad del comando more?***

· El comando no se ejecuta correctamente, ya que el fichero está comprimido, y aparece por tanto en binario.

· zless, sin embargo es un visor para archivos comprimidos, con éste si podemos observar el contenido del mismo.

· Con more, el contenido del fichero nos sale paginado (le vas dando al enter y se desplaza hacia abajo).

Q11. ***En el directorio /sbin/init.d observar los atributos del fichero at. ¿Se trata de un fichero binario o de texto? ¿Es ejecutable? ¿Podemos asegurar que en UNIX todos los ficheros ejecutables deben de ser necesariamente ficheros binarios? ¿Qué nombre reciben los ficheros ejecutables de tipo texto?.***

· El fichero at, es un fichero de texto y ejecutable, pero tan sólo para el root.

· No es necesario que en Unix sólo los ficheros binarios sean ejecutables, puesto que este no esta en binario, pero es ejecutable.

Los ficheros de texto ejecutables son conocidos como "scripts".

Q12. ***En el directorio /usr/bin, ejecutar el comando ls −l | less, ¿qué ocurre?¿Cuál es la funcionalidad del operador "|"?***

· El comando ls −l |less nos ejecuta un ls de forma paginada.

· El operador | sirve para encadenar comandos .

Q13. ***En el directorio (ejecutar el comando history. A la vista del resultado, ¿ Qué hace dicho comando? ¿qué ocurre si hacemos history | less?***

· El comando history, nos muestra un historial, con todas las acciones que hemos efectuado en la sesión.

· Con history | less, nos hace lo mismo que en el caso anterior, pero nos lo muestra paginado.

CONTINUACIÓN PRÁCTICA 2

Sistema Operativo Unix (continuación)

Ql. ***Copiar el fichero de texto vimrc del directorio /etc al directorio ~/usr/etc. Observar el contenido del fichero.***

cp /etc/vimrc (/usr/etc

vi vimrc

Q2. ***¿Qué hace el comando grep?. ¿Cuál es el resultado de ejecutar el comando grep set vimrc?.***

· El comando grep busca la palabra que le indiques dentro de los ficheros, es un filtro que busca coincidencias dentro del fichero.

· Si ejecutamos grep set vimrc, busca dentro del fichero de texto vimrc la palabra set.

Q3. ***¿Cuál es exactamente la funcionalidad de las opciones A, B, b, c, i, v?. Describir detalladamente el funcionamiento de los siguientes comandos:***

−A 3 Imprime la línea en la que aparece el término que le indiquemos y a su vez imprime las líneas que hayan por debajo. En este caso imprimiría 3 líneas por debajo de la línea en la que está lo que buscas con grep.

−B 4 Igual que −A pero éste te imprime las líneas que hallan por encima, en este caso 4 líneas.

−b Nos dice en que carácter se encuentra la palabra indicada en la línea.

−c Número de veces en la que aparece en el archivo el término indicado.

−1 Busca el término que le hallamos indicado sin distinguir mayúsculas y minúsculas.

−v Muestra las líneas donde no está el término.

grep −c set (/usr/etc/vimrc

número de veces que aparece 'set" en el fichero vimrc.

grep fi (/.profile

muestra todas las palabras con las letras "fi" que hay en el fichero, estén al principio, al final o incluso en medio de una plabra.

grep −c read (/profile

número de veces que aparece "read" en el fichero.

grep −ci read (/.profile

número de "read" que tiene el fichero indicado, sin tener en cuenta si son en mayúscula o en minúscula.

grep −v set (/usr/etc/vimrc

Imprime las líneas en las que no está la palabra "set".

grep −v set (/usr/etc/vimrc | less

Imprime las líneas en las que no está la palabra "set" y te las muestra de forma paginada.

Q4. ***¿Cómo deberíamos ejecutar el comando ls de tal forma que únicamente nos muestre los directorios?. Ejemplificarlo en el directorio /usr/sbin.***

Ls −F | grep /

· Como con ls −F asignas a los directorios el símbolo /, con el filtro grep solo te imprime los nombres que tengan en la línea a /, que son simplemente los directorios.

Q5. ***Describir detalladamente el funcionamiento de los siguientes comandos:***

history | grep ls

muestra las líneas donde has escrito anteriormente ls.

history | grep ls | less

muestra las líneas donde has escrito anteriormente ls y además te lo muestra paginado, si le das al enter va

bajando.

history | grep −c cd

cuenta las líneas con la que has escrito anteriormente con la palabra cd.

Q6. ***¿Cuál es la funcionalidad del comando whatis?. ¿Qué opciones admite?.***

· Es una base de datos que te dice que es lo que hace un comando especificado.

−d genera información de depurado.

−r interpreta cada palabra clave como regex.

−w las palabras claves contienen comodines.

−m incluye páginas de manual de otros sistemas.

−M las páginas de manual se buscarán en "ruta".

−v muestra la versión.

−h muestra este mensaje en uso.

Q7. ***Describir detalladamente el funcionamiento de los siguientes comandos:***

whatis cd Indica la funcionalidad del comando cd.

whatis ls Indica la funcionalidad del comando ls.

whatis −w mk* Indica la funcionalidad de todos los que empiecen por k.

whatis −w *mk * | less Los muestra con capacidad de desplazamiento.

Q8. ***Al ejecutar whatis mkdir aparecen dos secciones de manual distintas: ¿cuál es la sección que el comando man toma por defecto?. ¿Qué debemos hacer que el comando man nos muestre la otra sección?.***

· Por defecto muestra la sección 1.

· Se pone: man (nº sección) mkdir = man 2 mkdir

Q9. ***¿Qué hace el comando whoami?. ¿Qué utilidad puede tener?***

· Whoami: Muestra el nombre del usuario actual.

· Utilidad: Como Linux permite trabajar con diferentes usuarios, así podemos saber que usuario eres y si tienes acceso a determinadas acciones.

Q10. ***¿Qué hace el comando whereis?. ¿Cuál es exactamente la funcionalidad de las opciones b, m, s?. El comando tar, ¿donde se ubica?, ¿tiene página de manual?.***

· Whereis: Muestra la ruta del comando binario (ejecutable) y la ruta del manual para dicho comando.

Opciones:

−b: busca solamente los archivos binarios.

−m: busca solamente secciones del manual.

−s: busca sólo en secciones del manual.

· Tar: Se ubica en "/bin/tar"

· Si tiene página en el manual: /usr/include/tar.h

PRACTICA 3

Editor de textos vim (vi iMproved)

Modo Edición:

i insert inicia el modo edición en la posición del cursor.

a append inicia el modo edición en la posición posterior del cursor.

A append inicia el modo edición al final de línea.

o open crea una nueva línea debajo del cursor. Se inicia el modo escritura.

O open crea una nueva línea arriba del cursor. Se inicia el modo escritura.

Modo Comando:

Desplazamiento

[ESC] sale del modo texto y se entra en modo comando.

k desplaza el cursor una línea hacia arriba.

j desplaza el cursor una línea hacia abajo.

l desplaza el cursor un carácter hacia la derecha.

h desplaza el cursor un carácter hacia la izquierda.

0 desplaza el cursor al principio de la línea.

$ desplaza el cursor al final de la línea.

gg desplaza el cursor al final del documento.

G desplaza el cursor al principio del documento.

NG desplaza el cursor a la línea N.

z<CR> la línea actual pasa a ser la primera línea visible.

z. la línea actual pasa a ser la última línea visible.

^F página adelante.

^B página atrás.

Guardar, salir

:q salir al shell (sólo si se han guardado los cambios).

:q! salir al shell (aunque no se hayan guardado los cambios).

:w guardar los cambios.

:wq salir al shell guardando los cambios.

.r <file> inserta el contenido del archivo <file> en la posición del cursor.

.r! {command} inserta la salida estándar del comando {command} en la posición del cursor.

Borrar

x elimina el carácter en el que está situado el cursor.

dd elimina la línea en la que está situado el cursor.

Ndd elimina N líneas hacia abajo empezando por la línea en la que está el cursor.

Copiar y pegar

yy copia la línea en la que está en cursor.

p pega el contenido del portapapeles debajo de la línea en la que está el cursor.

P pega el contenido del portapapeles arriba de la línea en la que está el cursor.

Búsqueda

/{patrón} Busca el texto {patrón}

n siguiente búsqueda (del mismo patrón) hacia adelante.

N siguiente búsqueda (del mismo patrón) hacia atrás.

Miscelania

.help ayuda

Ql. ***Ejecutar el comando vi prueba en el directorio ~/usr/doc. Escribir el siguiente documento:***

Este es el documento Linux Sound HOWTO. Debe ser entendido como una guía de referencia rápida que cubre todo lo que necesitas saber para instalar y configurar el soporte de sonido bajo Linux. Se contestan las preguntas más frecuentes sobre el sonido en Linux, además de incluir referencias a otras fuentes de información sobre múltiples características relativas a la generación de sonido y música por ordenador.

El alcance está limitado a los aspectos sobre las tarjetas de sonido relacionadas con Linux. Remítase a los otros documentos listados en la sección Referencias para información más general sobre tarjetas de sonido y generación de sonido y música por ordenador.

Versiones nuevas de este documento serán enviadas periódicamente al grupo de noticias comp.os.linux.answers. También serán enviadas a varios ftp anónimos que archivan este tipo de información, incluyendo ftp ://sunsite.unc.edu/pub/Linux/docs/HOWTO/.

Versiones de hipertexto de éste y otros HOWTOs están disponibles en múltiples sites, incluyendo http://sunsite.unc.edu/mdw/mdw.html

La mayoría de las distribuciones de Linux en formato CD−ROM incluyen los HOWTOs en el directorio /usr/doc/, aunque también los puedes comprar en formato impreso a muchos vendedores.

· *vi prueba ...*

Q2. ***Salvar el documento (:w) y salir del vi (:q). Comprobar mediante el comando ls que el fichero realmente existe. ¿Qué permisos tiene?. Probar de ejecutar los comandos cat prueba y less prueba respectivamente, ¿funcionan correctamente?.***

· Los permisos son: −rw −r− −−r.

· Sólo podemos leerlo (r), no podemos escribir (w) ni ejecutarlo (x).

· Cat prueba y Less prueba funcionan correctamente.

Q3. ***Ejecutar el comando vi otra_prueba en el directorio ~/usr/doc. Escribir el siguiente documento:***

vi otra_prueba ...

Este documento nos permitirá observar la

utilidad del comando :r

Dicho comando permite insertar dentro del vi

texto escrito en otro documento

Q4. ***Ejecutar vi prueba. Colocar el cursor en la línea 7. Ejecutar el comando :r ~/usr/doc/otraprueba. ¿Qué es lo que ocurre?. Salir sin guardar los cambios. Volver a entrar y colocar el cursor en la línea 3. Ejecutar el comando :r! ls −1, ¿qué es lo que ocurre?. Salir sin guardar los cambios (:q!).***

· Nos copia el contenido de "otra_prueba" debajo de la línea 7.

· Nos copia al fichero prueba el contenido del directorio en el que estemos con sus propiedades...

· Salimos sin guardar cambios con :q!

Q5. ***¿El comando :r! cat ~/usr/doc/otra_prueba es equivalente al comando :r ~/usr/doc/otra_prueba?, ¿ porqué?.***

· Si es equivalente porque con :r! necesitas un comando y con :r un archivo.

· El contenido que inserta como documento es el mismo que el visor cat visualiza en pantalla.

Q6. ***Eliminar las líneas insertadas en la cuestión anterior mediante el comando dd (también con Ndd, especificando el valor adecuado de N). Guardar los cambios (:w) situar el cursor en la línea 3 y ejecutar el comando yy . Situar el cursor en la línea 12 y ejecutar el comando p. ¿Qué es lo que ocurre?.***

· Copia la línea que tienes en el portapapeles en la línea posterior a la 12 (13).

Q7. ***Eliminar la línea insertada en la cuestión anterior (dd). Repetir la cuestión Q5 pero empleando esta vez el comando 3yy. ¿Qué es lo que ocurre?. ¿Cuál es la diferencia entre los comandos p y P?***

· 3yy: Copia en el portapapeles 3 líneas consecutivas a partir de donde esté el cursor.

· P: pega encima de la línea del cursor.

· p: pega debajo de la línea del cursor.

Q8. ***Ejecutar los comandos que se describen a continuación y explicar la función de cada uno de ellos:***

· :set number muestra el número de línea.

· :set noruler quita la numeración de la línea/carácter.

· :set list Pone un símbolo (dólar) cada vez que hayas pulsado intro.

· :set nonumber no muestra el número de línea

· :set nolist Quita el símbolo (dólar).

· :set ruler Pone la numeración de la línea / carácter.

Q9. ***Mediante el comando :help, determinar la funcionalidad de las siguientes opciones:***

· Expandtab: identifica el valor de la tabulación.

· Ignorecase: Ignorar el caso de búsqueda de patrones.

· Insertmode: Modo insertar (para escribir en el vi).

·

Q10. ***Editar el fichero ~/.vimrc (VIM ResourCes). Dicho fichero contiene la configuración inicial del editor vi. En cualquier parte del documento, añadir las siguientes líneas de texto:***

set number

Salir guardando los cambios y verificar que, a partir de ahora, aparecen siempre los números de línea. Explicar detalladamente todas las opciones que el editor de textos vi utiliza por defecto.

Set noautoindent : texto justificado.

Set showmatch : muestra matchin y brackets.

Set showmode : cambia el modo de insertar por sobrescribir.

Set end compatible : Cambia ed por is.

Set nocompatible : no utilizan el comando vi con los ficheros que no son compatibles.

Set magic : cambia caracteres especiales en búsqueda de patrones.

Set term = Linux −m : desactiva el modo color en la consola.

Set l_kb=^? : en lugar de borrar cambia por el código ASCII.

Set t_kD=^[[3~ : elimina cuando borras ANSI.

PRACTICA 4

Sistema Operativo Unix (continuación)

***Ql*. Crear los directorios ~/prac4/man y ~/prac4/doc. Copiar los ficheros Mail−HOWTO.gz y ModemHOWTO.gz del directorio ~/usr/doc/howto/en al directorio ~/prac4/doc.**

mkdir ~/prac4/man

mkdir ~/prac4/doc

cp /usr/doc/HOWTO/en/Mail.HOWTO.gz ~/prac4/doc

cp /usr/doc/HOWTO/Modem−HOWTO.gz ~/prac4/doc

Q2. ***¿Para qué sirve el comando chmod?. En el directorio ~/prac4/doc ejecutar los siguientes comandos:***

1− chmod g+w Mail−HOWTO.gz

2− chmod o+w Mail−HOWTO.gz

3− chmod g−w, o−w MaiI−HOWTO.gz

4− chmod u="rw", g=" rw", o="r" Modem−HOWTO.gz

5− chmod 644 Modem−HOWTO.gz

¿Tiene sentido ejecutar el comando chmod 755 Modem−HOWTO.gz?, ¿porqué?.

Nota: Utilizamos man chmod

· Sirve para cambiar los permisos de acceso de los archivos.

1− Da permiso de escritura (w) al grupo (group).

2− Da permiso de escritura (w) al resto (other).

3− Quita los permisos de escritura (w) al grupo y al resto.

4− Define los permisos de usuario (user), grupo y resto según lo que pongas, en este caso rw, rw y r respectivamente.

· No tiene sentido darle esos permisos al fichero, ya que el fichero Modem.HOWTO.gz está comprimido, por lo que no podremos leer ni escribir nada en el ya que está en binario.

Q3. ***Restablecer los permisos de ambos ficheros a su estado inicial (rw−r−−r−−). ¿Cuál es la funcionalidad de los comandos gzip y gunzip?. El comando gzip −d ¿es equivalente a gunzip?. En el directorio ~/prac4/doc ejecutar los siguientes comandos:***

gunzip Mail−HOWTO.gz

gzip −d Modem−HOWTO.gz

Observar el contenido de los ficheros. ¿Tiene sentido ahora ejecutar el comando chmod 755 ModemHOWTO?.

· La funcionalidad del comando gzip es comprimir archivos, mientras que la del gunzip es descomprimirlos.

· gzip −d sí es equivalente a gunzip, pues los dos tienen como finalidad descomprimir.

· Si, ya que es de texto, y ya está descomprimido.

Q4. ***En el directorio ~/prac4/scripts editar el fichero miprimerx (comando vi). Escribir el siguiente texto:***

echo "Muestra el contenido del directorio /home/laboratorio (incluyendo ocultos)"

echo

ls −am /home/laboratorio

Salir guardando los cambios (:wq). ¿Podemos ejecutar el script miprimerx?. ¿Qué debemos hacer para que el fichero pase a ser ejecutable?. Verificar que el script funciona correctamente.

· Editamos el fichero y lo guardamos al salir:

· No podemos ejecutarlo.

· Para que sea ejecutable: chmod 755 miprimerx ó chmod u+x miprimerx

· Comprobamos que funciona.

Q5. ***Nuevo contenido del fichero miprimerx***:

echo Muestra el contenido del directorio especificado por el usuario

ls $1

Salir guardando los cambios y ejecutar miprimerx /usr. ¿Qué valor ha tomado, en este caso, la variable $1?.

· Reeditamos miprimerx:

echo Muestra el contenido del directorio especificado por el usuario

ls $1

· Ejecutamos "miprimerx /usr": La variable $1 ha tomado el valor de /usr

Q6. ***Nuevo contenido del fichero miprimerx:***

it test −z"$1";;then

echo $0: uso incorrecto.

echo "Debe especificar el nombre completo de un archivo."

else

chmod u−x,g−x,o−x $1

ls −l $1

echo "El fichero $1 ya no es ejecutable"

echo "El comando se completo satisfactoriamente."

fi

Salir guardando los cambios y ejecutar miprimerx, ¿que es lo que ocurre?. Ejecutar ahora miprimerx ~/prac4/doc/Modem−HOWTO. Explicar detalladamente el funcionamiento del script.

Reeditamos miprimerx ...

· Muestra "Debe especificar el nombre completo de un archivo", es decir, el mensaje antes expuesto.

· Quita los permisos de ejecucón (x) de Modem−HOWTO.gz, muestra una lista de contenido del directorio prac4/scripts/Modem−HOWTO y muestra los mensajes del script.

Q7. ***Nuevo contenido del fichero miprimerx***:

if test −z"$1";;then

echo $0[opcion] archivo. $0 −−help muestra ayuda.

else

case "$1" ¡n

−−help)

echo Opciones

echo help: pantalla de ayuda.

echo c: comprime el archivo.

echo 1: muestra el contenido del archivo.

echo b: borra el archivo.

;;

−c)

gzip $2

echo el archivo $2 ha sido comprimido satisfactoriamente.

;;

−1)

less $2

−b)

rm −rf $2

echo el archivo $2 se ha borrado completamente.

esac

fi

Salir guardando los cambios y ejecutar miprimerx, ¿que es lo que ocurre?. Ejecutar:

Al ejecutar miprimerx nos da una especie de ayuda que nos dice que faltan datos (debemos especificar las opciones del script).

1. miprimerx | ~/prac4/doc/Modem* ¿funciona?

2. miprimerx o ~/prac4/doc/Modem* ¿funciona?

3. miprimerx b ~/prac4/doc/Modem* ¿funciona?

1) Visualiza el contenido del fichero.

2) Comprime los datos.

3) Borra los anteriores citados.

Q8. ***Ejecutar vi miprimerx en el directorio ~/prac4/man y escribir el siguiente texto:***

.TH MIPRIMERX 1 "2 November 1999" "SuSE Linux 6.2" "Prueba Alumnos UPCT"

.SH NAME

miprimerx \− Permite ejecutar varios comandos empleando un único script

.SH SYNOPSIS

.B miprimerx

[opcion] archivo

SH DESCRIPTION

.1 miprimerx

es un script que permite borrar, ver o comprimir el archivo especificado. En

realidad, se trata de un script de prueba escrito por los estudiantes de la

UPCT (telemática) por lo que

.1 no creemos

que tenga mucha repercusion en la sociedad cientifica mundial.

.PP

De cualquier forma, describiremos detalladamente cada una de sus opciones.

.SH OPTIONS

.TP

.1 "\−−help"

Muestra la pantalla de ayuda.

.TP

.1 "\c"

Comprime el archivo.

.TP

.1 "\−l"

Muestra el contenido del archivo.

.TP

.l "\−b"

Borra el archivo

.SH SEE ALSO

.BR rm (1),

.BR gzip (1),

.BR less (1).

Salir guardando los cambios. Ejecutar el comando export MANPATH=$MANPATH: ~/prac4/man. ¿Que ocurre si ejecutamos man miprimerx'?. Si comprimimos el archivo ~/prac4/man/miprimerx, ¿sigue funcionando el comando man?.

· Al ejecutar el comando man miprimerx visualiza las características del fichero creado.

· Si comprimimos el fichero el comando no funciona.

Q9. ***En /usr/man/man1 se almacenan todas las páginas de manual pertenecientes a la seccionl. Mediante el comando zless observar una página cualquiera e identificar los campos más importantes. (la página de manual del comando man, explica detalladamente todas los campos existentes).***

· Las páginas están formadas por una descripción del comando junto con las respectivas opciones del mismo.

PRACTICA 5

Sistema Operativo Unix (continuación)

Ql. ***En el directorio ~/prac5/doc ejecutar el siguiente comando:***

ln −s /usr/doc/howto/en/Mail−HOWTO.gz email

¿Para qué sirve exactamente el comando ln?. ¿Podemos trabajar con el enlace simbólico "email" como si se tratase de un archivo normal?. Justificar detalladamente la respuesta.

· Crea uniones (enlaces) entre ficheros.

· Podemos trabajar con él, dado que es un enlace directo del fichero Mail−HOWTO.gz, y lo llama email.

Q2. ***¿Qué ocurre si ejecutamos ln −s /usr/doc/howto/en/Mail−HOWTO.gz?***

· Ocurre lo mismo que en Q1, pero no le cambia el nombre al enlace, se llama igual que el fichero.

Q3. ***Crear el directorio ~/prac5/bin. Copiar en ~/prac5/bin todos los archivos del directorio /usr/bin que empiezen por a. El comando cp, ¿ha respetado los enlaces simbólicos originales?. ¿Cómo deberíamos ejecutar el comando cp si quisiéramos mantener la integridad de los enlaces simbólicos?.***

md ~/prac5/bin

cp /usr/bin/a* ~/prac5/bin

· Cp no ha respetado los enlaces simbólicos.

· Para mantener los enlaces simbólicos se elige la opción −d de cp (cp −d).

Q4. ***Desde el directorio ~ ejecutar el comando tar −cvf miprimertar.tar prac5. ¿Qué hace exactamente el comando tar?, ¿Para qué sirven las opciones c, v, f?***

· Hace una copia de la estructura del directorio y sus ficheros.

−c crea un nuevo archivo.

−v muestra una lista de los archivos procesados.

−f le da un nombre al fichero (el que escribas después).

Q5. ***Borrar el directorio prac5. Ejecutar el comando tar −xf miprimertar.tar. ¿Qué ha ocurrido?. Borrar el archivo miprimertar.tar y ejecutar seguidamente el comando tar −czf miprimertar.tar.gz prac5. ¿Cuál es la diferencia con respecto a la cuestión Q4?. ¿Vale la pena emplear la opción z?***

· Lo descomprime y crea la secuencia guardada.

· La diferencia es que no muestra la lista de los archivos procesados y además lo comprime.

· No vale la pena pues la z (compresión) lo comprime, pero sin la z también lo hace.

Q6. ***¿Como podemos observar el contenido de un fichero generado por el comando tar sin tener que restaurarlo?. Ejemplificarlo con el archivo miprimertar.tar.gz***

· tar −ztvf miprimertar.tar.gz

Q7. ***En el directorio ~, editar el fichero .alias y escribir en él el siguiente texto:***

alias lsd="ls −aF | grep /"

alias dir='ls"

alias ll="ls −1"

alias la="ls −a"

Salvar el documento y cerrar todos los shells. Abrir una nueva cónsola, ¿qué comandos se ejecutan cuando escribimos lsd, dir, II y la respectivamente?. En consecuencia, ¿para qué sirve el comando alias?.

· lsd ejecuta ls −aF | grep / (Nos muestra sólo los directorios)

· dir ejecuta ls.

· ll ejecuta ls −l

· la ejecuta ls −a

· "alias" sirve para dar equivalencias entre comandos.

Q8. ***PS1 es una variable de entorno. Su valor especifica el mensaje que el shell mostrará en la línea de comando. Como ejemplo, ejecutar:***

export PS1 ="buenas tardes"

a) El prompt es ahora "buenas tardes".

b) Sale el nombre del laboratorio y el puesto del ordenador.

c) Aparece el usuario, el hostname y el directorio de trabajo.

Q9. ***Determinar la funcionalidad de las siguientes cadenas de caracteres aplicados a la variable de entorno PS1: \d, \h, \t, \T, \u, \w y \W.***

Sugerencia: man bash, sección PROMPTING.

\d aparece la fecha (día, mes, nº del día)

\h aparece el hostname hasta el "."

\t aparece la hora (hora, minutos, segundos) en 24h

\T aparece la hora (hora, minutos, segundos) en 12h

\u aparece el usuario

\w aparece el directorio de trabajo (desde el raíz)

\W aparece el directorio actual de trabajo

Q10. ***El comando set muestra el valor de todas las variables de entorno. ¿Existe una variable denominada PATH?, dicha variable, ¿incluye el directorio ~/usr/bin?***

· Si existe PATH, pero no incluye el directorio ~/usr/bin.

Q11. ***En el directorio ~/usr/bin crear un script muy sencillo que se denomine mi2script. Darle permisos de ejecución (chmod). Trasladarse al directorio ~ y ejecutar mi2script, a) ¿funciona?. Ejecutar export PATH=$PATH:~/usr/bin. Intentar de nuevo ejecutar mi2script, b) ¿funciona?. En consecuencia, c) ¿qué indica la variable de entorno PATH?.***

a) No funciona dentro de ~.

b) Ahora si funciona.

c) Indica ~/usr/bin y toma como raíz este directorio.

Q12. ***¿Que valor toma ahora dicha variable?.***

Sugerencia: set | grep PATH

· Aparecen las anteriores (que no indica ~/usr/bin) y ahora también ésta.

PRACTICA 6

Sistema Operativo Unix (continuación)

Ql. ***Crear el directorio ~/prac6. Editar el fichero miprimerprog.c y escribir en él el siguiente texto:***

```
#include <stdio.h>

int main ()

{

puts ("\nHello Word\n">;

exit ();

}
```

Guardar el fichero y en la línea de comandos ejecutar los siguientes comandos:

1º− gcc −c miprimerprog.c −Wall −pedantic => proceso de compilación.

2º− gcc miprimerprog.o −o miprimerprog => proceso de lincado.

Mediante el comando ls, observar los ficheros generados por el compilador. ¿Cuál es el fichero fuente, el objeto y el ejecutable?.

mkdir (/prac6

vi miprimerprog.c

· Compilamos y lincamos el código ...

gcc −c miprimerprog.c

gcc iniprimnerprog.o −o miprimerprog

· El fichero fuente es miprimerprog.c, el objeto miprimerprog.o y el ejecutable miprimerprog .

Q2. ***Ejecutar miprimerprog. A la vista del resultado de la ejecución, ¿qué hace exactamente la función puts?***

· Puts: Al ejecutar miprimerprog aparecen las palabras indicadas entre paréntesis y entre comillas después de puts.

Nota: \n añade una línea en blanco (retorno de carro).

\t añade una tabulación.

Todo lo que esté entre comillas aparece en pantalla.

Q3. ***Editar el fichero tipos.c y escribir en él el siguiente programa:***

```
#include <stdio.h>

int main ()

{

int ai, bi, ci;

double ad, bd, cd;

ai=10; bi=5; ci= ai+bi;

ad=10.2; bd=2.3; cd=ad/bd;

puts ("interger operation:\n");

printf ("\ta=%d, b=%d and c=a+b\n", ai, bi);

printf ("\tthe result is c=%d\n\n", ci);

puts ("float operation:\n");

printf ("\ta=%f, b=%f and c=a/b\n", ad, bd);

printf ("\tthe result is c=%f\n\n", cd);
```

exit (0);

}

Salvar el documento. Compilar y lincar el código escrito (nombrarlo tipos). Ejecutarlo y describir detalladamente la funcíonalidad de todas y cada una de las instrucciones que conforman el programa.

#include <stdio.h> −Para poder usar los comandos que necesitamos, entre ellos puts y printf

int main ( ) −Siempre se empieza con main y dentro de { } el programa

{

int ai, bi, ci; −i: variable de tipo entero

double ad, bd, cd; −d: variable de tipo decimal

ai=10; bi=5; ci= ai+bi; −En estas dos líneas se introducen valores a las

ad=10.2; bd=2.3; cd=ad/bd; variables

puts ("interger operation:\n");

printf ("\ta=%d, b=%d and c=a+b\n", ai, bi); −printf actúa igual que puts pero puedes

printf ("\tthe result is c=%d\n\n", ci); utilizar variables

puts ("float operation:\n");

printf ("\ta=%f, b=%f and c=a/b\n", ad, bd); (*)

printf ("\tthe result is c=%f\n\n", cd);

exit (0); −Significa que termino el programa

}

%d: Imprime variable de tipo entera (int)

%f: Imprime variable de tipo decimal (double)

(*): Representa en pantalla los valores dados entre comillas tomando los valores enteros ai, bi representados por %d y %f respectivamente.

Q4. Dados dos números enteros cualesquiera, comprobar si son divisibles entre si:

Si: "los dos números son divisibles"

No: "los dos números no son divisibles"

#include <studio.h>

```
int main ( )
{
int a, b, c, R;
a=15; b=3; c=a/b; R=a−c*b;
if (R==0)
{
printf ("Son divisibles\n");
}
else
{
printf ("No son divisibles\n")
}
exit (0)
}
```

Q5*. Determinar la funcionalidad del siguiente programa.*

```
#include <stdio.h>
main()
{
int i, c, R;
for (i =1; i <= 100; i = i +1)
{
c = i/3;
R = i − 3*c;
If(R = = 0) printf("%d\n", i);
}
}
```

La sentencia for da valor a las variables, i va desde 1 hasta un número siempre menor o igual a 100. Cuando el valor de R es igual a 0 se imprime en pantalla el valor de i, y para que esto ocurra, c debe ser entero, es decir múltiplo de tres, estos son los valores que apareceran en pantalla.

Q6. Simplificar el programa anterior (sugerencia. las variables c y R no son estrictamente necesarias).

```
#include <stdio.h>

main()

{

int i;

for (i = 0; i <= 100; i = i + 3)

{

printf("%d\n", i)

}

}
```

Q7. ***Modificar el programa anterior de forma que el resultado salga impreso por pantalla en 4 columnas.***

```
#include <stdio.h>

main()

{

int i;

int col;

col = 1;

for (i = 0; i <= 100; i = i + 3)

{

printf("%d\t", i);

col = col + 1

íf(col = = 5)

{

print ("'\n");
```

col =1

}

}

PRÁCTICAS 7 Y 8

"INSTALACIÓN Y RECOMPILADO DEL NÚCLEO DE LINUX"

PRACTICA 9

Lenguaje de programación C

Q1. ***Editar el programa (llamarlo prac9.c), compilarlo y lincarlo. ¿Qué hace exactamente dicho programa?, ¿cuáles son las diferencias entre las funciones príntf y puts?, ¿para qué sirven las instrucciones for, switch e if?, ¿cuál es exactamente la sintaxis de cada una de estas instrucciones?. Describir detalladamente el algoritmo y la funcionalidad de cada una de las variables.***

#include <stdio.h>

main ()

int dia, mes, maxdias;

int col, i;

col = 6;

for (mes = 1; mes <= 12; rnes = mes + 1)

{

switch (mes)

{

case 1: puts("Enero");

maxdias = 31;

break;

case 2: puts("Febrero");

maxdias = 29;

break;

case 3: puts("Marzo");

maxdias = 31;

```
break;
case 4: puts("Abril");
maxdias = 30;
break;
case 5: puts("Mayo");
maxdias = 31;
break;
case 6: puts("Junio");
maxdias = 30;
break;
case 7: puts("Julio");
maxdias = 31;
break;
case 8: puts("Agosto");
maxdias = 31;
break;
case 9: puts("Septierrbre");
maxdias = 30;
break;
case 10: puts("Octubre");
maxdias = 31;
break;
case 11: puts("Noviembre");
maxdias = 30;
break;
case 12: puts("Diciembre");
```

```
maxdias = 31;

break;

}

puts("Lun\tMar\tMier\tJuev\tVier\tSab\tDom");

for (i = 1; i < col; i = i + 1) printf("\t");

for (dia = 1; dia <= maxdias; dia = dia + 1)

{

printf("%d\t", dia);

col = col + 1;

if (col == 8)

{

printf("\n");

col = 1;

}

}

printf("\n\n");

}

}
```

Este programa crea un calendario del año 2000 al ejecutarlo.

La diferencia entre printf y puts es que printf imprime en pantalla el valor de una variable, mientras que la puts imprime lo que se le diga (no sólo variables).

If: comando de condición (si ...).

For: comando que da valor a las variables (para...).

Switch: comando que indica la variable que cambiará de valor.

Q2. ***¿Que ocurrirá si ejecutamos prac9 | grep −A6 Agosto?***

Busca e imprime en pantalla el mes de agosto en el fichero prac9.

Q3. ***Sabiendo que el 1 de enero del año 2000 fue sábado y que todos los años múltiplos de 4 son bisiestos,***

escribir un programa que determine el día de la semana del 1 de enero de cualquier año del siglo XXI.

PRÁCTICA 10

Servicios de Internet

En esta práctica se estudiarán los siguientes servicios de internet: correo electrónico (email), navegador de páginas web, talk, file transport protocol (ftp) y telnet.

Q1. ***lynx es un navegador de páginas web muy sencillo, que sólo admite caracteres ASCII. Debe ejecutarse necesariamente a través de una consola Unix.***

Opciones de lynx:

g (go) navegar a través de la página web especificada.

q (quit) salir de¡ programa.

(un paso hacia atrás.

® un paso hacia adelante.

(siguiente enlace.

(enlace anterior.

<space> siguiente página.

A través de un buscador (yahoo, excite, ole, etc.) determinar la página web del periódico "El Mundo". Conectarse a dicho periódico y buscar el artículo publicado en la sección denominada "Ultima" cuyo autor es Francisco Umbral.

· La página es "http://www.el−mundo.es"

Q2. ***Otra forma de navegar es a través de un navegador gráfico, como por ejemplo netscape. La presentación de la información es mucho más agradable que la proporcionada por lynx, aunque algunas veces resulte un poco más lento.***

A través de un buscador, determinar la página web del periódico "ABC". Conectarse a dicho periódico y buscar el artículo publicado en la sección denominada "opinión" cuyo autor es Jaime Campmany.

· La página es "http://www.abc.es".

· Con 'Edit'−'find in the page' buscamos el artículo, que se titula "La lluvia".

Q3. ***Escribir el siguiente texto mediante el editor vi (nombre del fichero: consejos)***

Hola.

Espero que ya te hayas repuesto de la resaca del jueves por la tarde. Nosotros, sin embargo, preferimos aprovechar la festividad de San Jose para estudiar fundamentos de los computadores.

Hasta pronto...

Salvar el documento. Ejecutar el comando cat consejos | mail −s 'paellada'fc0XX@it5−XX.upct.es". Automáticamente, se envía por correo electrónico el mensaje escrito al usuario fc0XX conectado a la máquina it5−XX.

cat −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−− visualizador

consejos−−−−−−−−−−−−−−−−−−−−−−−−−−−− archivo a mandar

| mail −s −−−−−−−−−−−−−−−−−−−−−−−−−−−− comando con el que se envían los mensajes

'paellada'−−−−−−−−−−−−−−−−−−−−−−−−−−− título del mensaje

fc0XX@it5−XX.upct.es−−−−−−−−−−− dirección a la que enviamos el mensaje

(nº usuario) (nº PC)

Q4. ***Para leer y escribir correos electrónicos, podemos utilizar el propio netscape:***

Edit, Preferences...

Mail & Newsgroups: Mail servers. Seleccionar "pop" y eliminarlo pulsando el botón "delete". Pulsar el botón "add". En server type especificar MoveMail. Finalmente, escribir el nombre del grupo de trabajo (fcOXX) en el campo User Narne. Mail & Newsgroups: Identity. Escribir vuestra dirección de correo electrónico en el campo Email Address (fcOXX @ it5−XX.upct.es).

A partir de este momento, podréis enviar y recibir emails. Verificar que habéis recibido el email escrito anteriormente por alguno de vuestros compañeros. Enviadle una respuesta.

・Pulsar 'Get msg' para ver los mensajes.

・Para contestar 'Reply' y escribimos el texto a enviar.

Q5. ***File Transport Protocol (ftp) es una aplicación que permite enviar y recibir ficheros.***

Los comandos básicos que acepta el programa son:

ls: ver el contenido del directorio remoto.

cd: cambiar el directorio del ordenador remoto.

!: Ejecutar cualquier comando unix en local.

quit: Salir de programa.

put <fichero>: transferir el archivo <fichero> del ordenador local al remoto.

get <fichero>: transferir el archivo <fichero> del ordenador remoto al local.

mput: transferencia múltiple (con caracteres wildcard).

mget: transferencia múltiple (con caracteres wildcard).

Conectarse a ftp.rediris.es (login: anonymous, password: vuestra dirección de email). Buscar el archivo elm−2.4p125pgp3. tgz y copiarlo en directorio ~/elm.

Nota: Normalmente, el fichero denominado ls−lR.gz incluye una referencia a todos los archivos ubicados en el servidor ftp. Con zless podemos buscar el directorio donde se encuentra un determinado archivo.

· Para conectarse al ftp se hace lo siguiente: ftp ftp.readiris.es.

· Para descargar el fichero se hace "get ls−lR.gz".

· Se visualizar mediante el comando zless y se introducir la sentencia para buscar lo que nos interesa "\elm−2.4pl", pues al fichero es muy grande.

Situación: ".\software\linux\sunsite\system\mail\mua\"

Q6. ***Repetir el apartado anterior pero utilizando netscape como cliente ftp. En este caso, el fichero a transferir deberá ser Mail−HOWTO.***

· Es lo mismo que el caso anterior, pero ahora no se utiliza el comando "get", sino que se hace un simple click en el fichero escogido y este se descarga.

Situación:".\sotware\linux\distribuciones\redhat\redhat\oldrelease\red6.0\alpha\doc\HOWTO\"

Q7. ***También podemos transferir ficheros a través del correo electrónico. Abrir el navegador y enviar un correo al compañero que comparte la misma mesa que incluya el fichero elm−2.4pI25pgp3.tgz (a través de la opción attach).***

· Para transferir un fichero al mensaje se escoge attach en el cuadro de creación de éste, después se escoge el fichero a adjuntar y se envía.

Q8. ***talk es una aplicación que permite que dos usuarios conectados en Internet puedan "hablar" electrónicamente. Su sintaxis es: talk <usuario>@ <ordenador>.***

Verificar el funcionamiento de talk hablando con el compañero ubicado en el ordenador contiguo.

· Utilización: talk fc007@it5−03

Q9. ***Finalmente, la aplicación telnet permite conectarse a una máquina remota y trabajar en ella como si estuviéramos en local, su sintaxis es: telnet <ordenador>.***

Verificar el funcionamiento de telnet conectándose al ordenador remoto situado en la misma mesa.

· Para la utilización del programa telnet necesitaremos saber la contraseña de la máquina remota a la que vamos a acceder.

Utilización: telnet it5−03

PRACTICA 11

AWK

Awk es un lenguaje de programación que permite procesar ficheros de texto. Se le denomina awk en honor a los programadores que lo definieron: Alfred V. Aho, Peter J. Weinberger y Brian W. Kernighan.

Q1. ***Verificar que en el directorio ~/prac11 existen 2 ficheros denominados "asignaturas" y "telefonos" respectivamente. ¿Son ficheros de texto o binarios?. ¿Cómo podemos observar su contenido?.***

· Observando el contenido de los ficheros con un visualizador como less, cat, ... vemos que son archivos de texto.

El fichero "telefonos" contiene información sobre los profesores pertenecientes a las áreas de Telemática (it), Teoría de la Señal y Comunicaciones (tsc) y Lenguajes y Sistemas Informáticos (¡si). Cada línea de este fichero recibe el nombre de registro; a su vez, cada registro está compuesto por campos (por lo general, separados por espacios en blanco aunque, en este caso, el separador de campos es el carácter':').

Q2. ***¿Qué tipos de campos conforman cada uno de los registros del fichero "telefonos"?***

· Los campos, separados por ":" son:

Campo 1 ($1) Campo 2 ($2) Campo 3 ($3)

Área : nombre y apellidos : teléfono

La línea entera (los 3 campos) se llama registro.

Q3. ***Describir detalladamente la funcionalidad de los siguientes comandos:***

· awk −F: '{print $2}' telefonos

Muestra el segundo campo del archivo telefonos.

· awk −F: '{print $2,$3}' telefonos

Muestra el segundo y tercer campo del archivo telefonos.

· awk −F: '{print $2", Te1f.:"$3}' telefonos

Muestra el segundo y tercer campo del archivo telefonos, pero al campo 3 le pone el título Telf.:. Por ejemplo Telf.:5590

Q4. ***Awk permite filtrar aquellos registros que no satisfacen una determinada condición. ¿Qué hacen exactamente los siguientes comandos?***

· awk −F: '/tsc/ {print $2}' telefonos

Muestra el campo 2 de la categoría tsc ($1=tsc) de telefonos.

· awk −F: '/53/ {print $2,$3}'telefonos

Muestra el campo 2 y 3 (nombre y teléfonos) de todos los registros que contengan un 53.

· awk −F: '/M/ {print $2,$3}' telefonos

Muestra el campo 2 y 3 (nombre y teléfonos) de todos los registros que contengan una M.

· awk −F: '/Alejandro/ {print $2, $3}' telefonos

Muestra el nombre y el telefono de los registros que contengan Alejandro (en este caso solo aparece uno).

Nota: /it/ busca it en un registro entero.

{$1= = "it"} busca it solo en el campo 1, no en todo el registro.

Q5. ***Otra forma más compleja (aunque mucho mas versátil) de filtrar registros es a través de condiciones lógicas. Los operadores lógicos definidos en awk son: == (igual a), != (diferente a), && (and lógica) y || (or lógica). Describir detalladamente la funcionalidad de los siguientes comandos:***

· awk −F: '$1= ="it" || $1 = ="ls" {print $2", area de conocimiento = "$1}' telefonos

Busca todos los que contengan it y lsi en el campo 1, y te imprime el campo 2 (nombres) y añade un título ("area de conocimiento") al campo 1 del archivo telefonos.

· awk −F: '$3 >= 5300 && $3 < 5400 {print $2, $3}'telefonos

Muestra los números de teléfono y sus nombres (campo 3 y 2) cuyos números de teléfono sean mayores o iguales que 5300 y menores que 5400.

· awk −F: '$2= ="Malgosa Sanahuja Josemaria" {print $3}'telefonos

Muestra el teléfono del registro en cuyo campo 2 (nombres) este "Malgosa Sanahuja Josemaria" (solo hay un registro que lo contengan, pues el nombre no se repite).

Cualquier programa escrito en awk tiene la siguiente estructura:

BEGIN {instrucciones}

/patrón de búsqueda/ (o también, una condición lógica)

{instrucciones}

END {instrucciones}

Las instrucciones encerradas dentro del bloque BEGIN se ejecutan ANTES de empezar a leer el fichero de texto. Después, para cada registro que satisfaga el patrón de búsqueda, se ejecutarán las instrucciones escritas entre las llaves principales. Finalmente, al terminar la lectura del fichero de texto, se ejecutan todas las instrucciones encerradas dentro del bloque END.

Q6. ***Describir detalladamente el funcionamiento de los siguientes comandos:***

· awk −F: BEGIN {num_reg=0} {num_reg = nurnreg +1} END {print num_reg}' telefonos

Empieza desde el registro 0 y va sumando hasta el último registro, y nos imprime el número total de registros (END).

· awk −F: BEGIN {print "Listado de telefonos del area TSC:"} /tsc/ {print $2,$3}' telefonos

Crea una lista de teléfonos y nombres pertenecientes al campo 1 tsc. Al principio de la lista imprime el título "Listado de telefonos del area TSC".

· awk −F: BEGIN {print "Componentes del Area de Ingenieria Te1ematica:"} /it/ {print $2}'telefonos

Crea una lista de nombres pertenecientes a it, con el título "Componentes del Area de Ingenieria de Telemática:" al principio de la misma.

· awk −F: '$1 = ="lsi" && $3 >= 5300 && $3 < 5400 {print $2} END {print "\nMienbros de LSI en Marina}' telefonos

Muestra los nombres en cuyo campo esta lsi y cuyos números de teléfono sean mayores o iguales que 5300 y menores que 5400, al final te muestra un título que dice: "Miembros de la LSI en Marina" (pero con una línea en blanco por /n).

Q7. ***Sino se especifica ningún fichero, awk toma como fichero de entrada la entrada standard. Ello permite ejecutar awk a través de pipes (mediante el carácter '|'). Ejecutar los siguientes comandos y explicar detalladamente la funcionalidad de cada uno de ellos:***

· ls−la

Muestra el contenido, incluido ocultos, con los atributos, del ubicaje actual.

drwxr−xr−x 2 fc007 alumnos 1024 mar 24 19:56 alumnos

Lo consideramos un registro con 9 campos ($1,$2,$3,$4,$5,$6,$7,$8,$9).

· ls −la | awk'{print $5, $9}'

Te imprime el nombre de los archivos ($9) y su tamaño ($5) (bytes) dentro del directorio donde estamos (prac11).

· ls −la | awk 'BEGIN {x=O} {x = x + $5} END {print "Bytes usados = "x}'

Te suma los bytes ocupados por los 4 archivos que están en prac11/, y te lo imprime con el título: "Bytes usados".

· ls −la | awk 'BEGIN {x=O} $9 !="." && $9 !=".."{x = x + $5} END {print "Bytes usados = "x}'

Te suma los bytes ocupados por los archivos que no sean ocultos (. y ..) y que estén en prac11/, y te lo imprime con el título: "Bytes usados".

· ls −la | awk 'BEGIN {x=O} {x = x + $5} END {if (x > 10000) print "Ocupas demasiado disco"}'

Te suma todos los archivos (lo que ocupan en bytes), y si supera los 10000 bytes, te imprime "Ocupa demasiado disco".

El fichero "asignaturas" contiene información sobre algunas de las asignaturas impartidas por las áreas de IT, TSC y LSI. Su estructura es la siguiente:

Técnica o Superior (t y s respectivamente). Curso en que se imparte (1, 2, 3, 4, 5). Nombre de la asignatura. Créditos totales. Cuatrimestre en que se imparte (1, 2, a).

Q8. ***Empleando únicamente el comando awk, contestar a las siguientes preguntas:***

¿Cuántas asignaturas imparten las áreas de IT, TSC y LSI? ¿Qué asignaturas se impartirán en 4 de carrera? ¿Qué asignaturas se impartirán en 2 de carrera (tanto técnica como superior)? ¿Cuantos créditos se imparten en total en la escuela técnica?, ¿y en la superior? ¿Cuantos créditos se imparten durante el 2 cuatrimestre de la técnica? ¿Qué asignaturas de la técnica son anuales? ¿Cuantos créditos de la superior son anuales?

· awk −F: 'BEGIN{num_reg = 0} {num_reg = num_reg + 1} END {print num_reg}' asignaturas

41

· awk −F: '$2 = = 4 {print $3}' asignaturas

Arquitectura de Computadores/ Redes De Ordenadores

· awk −F: '$2 = = 2{print $3}' asignaturas

Fundamentos De Programación/Conmutación/Software De Com/.../Sist. Lineales/Sist. Elect. Digitales=12

· awk −F: 'BEGIN {x = 0} $1 = = "t" {x = x + $4} END {print x}' asignaturas

114

· awk −F: 'BEGIN {x = 0} $1 = = "s" {x = x + $4} END {print x}' asignaturas

139

· awk −F: '$5 = = "a" {print $3}' asignaturas

Fundamentos De Computadores/ Fundamentos De Programación/Redes Y Servicios De Com./ Redes De Ordenadores/ Tratamiento Digital de la Señal

· awk −F: 'BEGIN {x=0} $1 = = "s" && $5 = = "a" {x = x + $4} END {print x}' asignaturas

28'5

PRACTICA 12

AWK (continuación)

Q1. ***Las directivas '>' y '>>' permiten redireccionar la salida standard (que por defecto es la pantalla) hacia cualquier otro dispositivo. Situarse en el directorio ~/prac12 y ejecutar los siguientes comandos:***

ls −la /var history ls −la /var > prueba−ls.txt

history > prueba−hist.txt

¿Se han creado los ficheros prueba−ls.txt y prueba−hist.txt?, ¿qué contenido tienen?. A la vista de los

resultados, ¿qué hace exactamente la directiva

· Se han creado los ficheros prueba_ls.txt, que contiene lo que te imprime en pantalla si haces ls −la /var; mientras que el fichero prueba_hist.txt contiene lo que te imprime en pantalla si haces history.

· > redirecciona la salida de un comando hacia un dispositivo especificado (en este caso un archivo de texto).

Q2. ***Ejecutar los siguientes comandos:***

ls −la / > prueba−ls.txt(observar el contenido del fichero prueba−is.txt)

history >> prueba−ls.txt (observar el contenido del fichero prueba−ls.txt)

A la vista de los resultados, ¿qué hace exactamente la directiva ""?

· Te envía al fichero prueba_ls.txt lo que te imprime en pantalla si haces ls −la / (directorio raíz).

· >> añade al archivo prueba_ls.txt la salida de history (sin borrar el contenido anterior).

Borrar todos los ficheros generados en los apartados anteriores (rm prueba). El fichero "asignaturas" contiene información sobre algunas de las asignaturas impartidas por el departamento de Tecnologías de la Información y Comunicaciones. Su estructura es la siguiente:

Campo 1: Técnica o Superior (t y s respectivamente).

Campo 2: Curso en que se imparte (1, 2, 3, 4, 5).

Campo 3: Nombre de la asignatura.

Campo 4: Número de créditos.

Campo 5: Cuatrimestre en que se imparte (1, 2, a).

Campo 6: ¿Optativa? (opt, nopt).

Q3. ***Se desea generar una nueva base de datos (llamada "asignaturas.q3) idéntica a la anterior, con la salvedad de que utilice el espacio en blanco como delimitador de campos (en vez del caracter':'). Para ello, ejecutar el siguiente comando:***

awk −F: '{print $1" "$2" "$3" "$4" "$5" "$6}' asignaturas > asignaturas.q3

Observar el contenido del fichero "asignaturas.q3, ¿cuál es ahora el separador de campos?. Si en adelante utilizamos este fichero, ¿deberemos emplear la opción −F: al invocar el comando awk?

· El separador de campos es ahora un espacio en blanco.

· No, pues −F: indica que el delimitador sea :, pero si es un espacio, no es necesario −F, pues lo toma por defecto.

El fichero "mas−asignaturas" tiene exactamente la misma estructura que "asignaturas" y contiene información sobre algunas asignaturas adicionales que no están incluidas en el fichero original.

Q4. ***Sustituir el delimitador de campo del fichero "mas_asignaturas" por un espacio en blanco (la nueva base de datos debe llamarse "mas_asignaturas.q4").***

· awk −F: '{print $1" "$2" "$3" "$4" "$5" "$6}' mas_asignaturas > mas_asignaturas.q4

Q5.***Ejecutar el siguiente comando y describir detalladamente su funcionalidad:***

cat mas_asignaturas.q4 " asignaturas.q3

· Te añade mas_asignaturas.q4 al fichero asignaturas.q3, pero sin borrar el contenido anterior.

Q6. ***Escribir, mediante el editor vi, el siguiente texto (guardar el archivo con el nombre "programa")***

BEGIN {

Cred_tec = 0;

Cred_sup = 0;}

{

if ($1 == "t") cred_tec = cred_tec + $4;

else cred_sup = cred_sup + $4;

END {

print "Total creditos en la tecnica = "cred_tec;

print "Total creditos en la superior ="cred_sup;}

Ejecutar el siguiente comando "awk −f programa asignaturas.q3". ¿Para qué sirve la opción −f?, ¿Qué hace exactamente este programa?

· La opción −f sirve para especificar el nombre del programa (nombre de lo escrito en vi).

· El programa suma los créditos de la técnica y la superior, y te los muestra con un título de esta forma:

Total creditos en la tecnica = 126

Total creditos en la superior = 223

Q7. ***Modificar el contenido del fichero "programa" con el siguiente texto:***

{

if ($6 "opt") x = "si";

else x = "no";

print $1" "$2" "$3" '$4,' '$5" "x;

}

Ejecutar el comando "awk −f programa asignaturas". ¿Qué hace exactamente este programa?. Crear una nueva base de datos (llamada "asignaturas.q7") tal que el campo 6 tome el valor "si" cuando la asignatura sea optativa y el valor "no" si la asignatura es obligatoria.

· Ahora el programa te informa si las asignaturas son optativas o no.

· Creamos una nueva base de datos:

awk −F: −f programa asignaturas> asignaturas.q7

PRACTICA 13

Scripts Avanzados

Q1. ***Mediante el editor de textos vi, escribir el siguiente script (nombre script1):***

for n in /bin /sbin /usr/bin /usr/sbin; do

cd $n

ls −l

done

¿Para qué sirve exactamente la instrucción for? ¿qué valores puede tomar la variable n?. En consecuencia, ¿qué hace exactamente el scrípt?.

Nota: una vez editado el script en vi, para hacer el archivo ejecutable debemos cambiarle los atributos (por ejemplo con chmod 755 script1).

· for es una condición para el caso de que se den unas variables, en este caso la variable n.

· n puede tomar como valores los distintos directorios /bin /sbin /usr/bin /usr/sbin.

· Para cada directorio $n ejecuta el comando ls −l, es decir, te imprime en pantalla el contenido de los directorios anteriores.

Q2. ***Mediante el editor de textos vi, escribir el siguiente scrípt (nombre script2):***

for n in Ibin /sbin /usr/bin /usr/sbin; do

cd $n

echo −n "Total kilobytes en $n: "

ls −l | awk 'BEGIN{x=0} {x=x+$5} END{print x/1000}'

done

¿Qué hace exactamente este script?, la instrucción "cd $n", ¿es estrictamente necesaria?.

· Ejecuta para cada directorio /bin /sbin /usr/bin /usr/sbin los comandos ls −l y awk, poniéndole el título "Total Kilobytes en $n (nombre del directorio que toma n)".

− awk te suma los bytes de cada fichero (dentro de los directorios directorios /bin /sbin /usr/bin /usr/sbin) y los divide entre 1000 para que sean Kilobytes.

− echo −n presenta el nombre del fichero y el tamaño en una misma línea, si no se pone aparece el nombre del fichero y una línea más abajo su tamaño.

· cd $n es necesario para decir en que directorio queremos sumar el tamaño de los ficheros.

Q3. ***Mediante el editor de textos vi, escribir el siguiente script (nombre script3):***

```
for n in /home/alumnos/fc*; do

if test −d $n;

then echo −n "Total kilobytes en $n:

ls −l | awk 'BEGIN{x=0} {x=x+$5} END{print x/1000}'

fi

done
```

¿Qué valores puede tomar la variable n?, ¿qué hace la opción −d delcomando test?.. En consecuencia, ¿qué hace exactamente el scrípt?

· n toma como valores todos los archivos y directorios que estén en /home/alumnos y empiecen por 'fc' (fc*).

· La opción −d del comando test implica que la variable n solo puede ser directorio.

· El script te va sumando, dentro de cada directorio que toma la variable n, el tamaño de los ficheros, y así te muestra el tamaño de cada directorio que empiece por fc y te pone el título "Total Kilobytes en $n".

Q4. ***En el directorio /(existe un fichero denominado passwd. Observar cuidadosamente su contenido y responder a continuación a las siguientes preguntas:***

¿Qué carácter utiliza para separar los distintos campos?

· ":"

¿Cuantos campos conforman un registro?

· 7 campos.

¿Qué indican los campos 1, 2, 3, 4 y 6?

Campo 1: nombre de usuario.

Campo 2: password cifrado.

Campo 3: user identifier (UID).

Campo 4: group identifier (GID).

Campo 5: descripción

Campo 6: directorio de usuario.

Campo 7: shell que emplea.

¿Tenemos permiso de lectura?

· Sí.

¿Y de escritura?

· Sí.

Q5. ***Mediante el editor de textos vi, escribir el siguiente script (nombre script4):***

for n in `awk −F: '$3>599 && $3<651 {print $6}' ~/passwd` do

if test −d $n; then

cd $n echo −n "Total kilobytes en $n: "

ls −l | awk 'BEGIN{x=0} {x=x+$5} END{print x/1000}'

fi

done

¿Que valores toma la variable n?. Con respecto al scrípt Q3, ¿qué mejoras aporta esta segunda versión?

· n toma como valores los números del campo3 (user identifier) que estén entre 600 y 650 dentro del fichero passwd.

· Si se desplaza el directorio de trabajo en el de Q3) dara error, mientras que este funcionará correctamente.

Q6. ***Dentro del directorio ~/prac13, existen a su vez otros 10 directorios denominados dírl, dír2, etc.) y un fichero tipo tar denominado prac.tar. Escribir un script (script5) que permita copiar y restaurar el fichero "prac.tar" en cada uno de los directorios.***

for n in ~/prac13/dir ; do*

if test −d $n ; then

*cp ~/prac13/prac.tgz $n /*copia*/*

*cd $n /*se va a los directorios*/*

*tar zxf prac.tgz /*descomprime*/*

*rm prac.tgz /*borra el archivo comprimido*/*

fi

done

PRACTICA 14

Anatomía de un disquete

En esta práctica se propone "investigar" el contenido de un disquete, a través del análisis de las distintas zonas que lo componen. Para ello, se dispone de la siguiente Información:

· Una imagen parcial del disquete en la que aparecen todas las zonas de interés para la práctica. A partir de ella, se pueden extraer las características básicas M disquete (desconocidas a priori por el alumno) como por ejemplo el tipo de disquete (360KB, 720KB, 1.22MB, 1.44MB), su formato, el nombre de volumen, el número de sectores por pista, el numero de caras, el numero de pistas, ficheros y directorios contenidos, los sectores utilizados, disponibles y defectuosos, etc.

· Un breve apéndice, donde se explica la conversión entre clusters y sectores.

· Un capítulo del libro titulado "El Universo Digital del IBM PC, AT y PS2" donde se explica detalladamente la estructura interna de un disquete.

En primer lugar, el alumno debe aprender la estructura de datos que utilizan los disquete (leyendo detenidamente el capítulo del libro). Una vez resuelto este primer paso, deben contestarse las siguientes cuestiones, todas ellas referidas a la imagen del disquete que se proporciona.

Q1. ***¿Qué indican los offsets 3, 11, 13, 14, 16, 17, 19, 21. 22, 24, 26 y 28 (sugerencia: ver la tabla "Formato del sector de arranque")?. Representar en una tabla similar los valores que toman en la imagen de nuestro disquete.***

¿QUÉ INDICA?

− Offset 3, formado por 8 bytes, indica la identificación del sistema, en nuestro caso MSDOS5.0

− Offset 11, formado por 1 palabra, indica los bytes por sector, en nuestro caso 0200 (512)

− Offset 13, formado por 1 byte, indica los sectores por cluster , en nuestro caso 01 (1)

− Offset 14, formado por 1 palabra, indica los Sectores reservados al principio, en nuestro caso 0001 (1)

− Offset 16, formado por 1 byte, indica el Nº de copias de la FAT, en nuestro caso 02 (2)

− Offset 17, formado por 1 palabra, indica el Nº de entradas al directorio raíz, en nuestro caso 00e0 (224)

− Offset 19, formado por una palabra, indica el Nº sectores de disco, en nuestro caso 0b40 (2880)

− Offset 21, formado por 1 byte, indica el Byte de tipo de disco, en nuestro caso f0

− Offset 22, formado por 1 palabra, indica el Nº de sect. ocupados por cada FAT, en nuestro caso 0009 (9)

− Offset 24, formado por 1 palabra, indica el Nº de sectores por pista, en nuestro caso 0012 (18)

− Offset 26, formado por 1 palabra, indica el Nº de cabezas, en nuestro caso 0002 (2)

− Offset 28, formado por 2 palabras, indica el Nº de especiales reservados, en nuestro caso 0000 0000 (0)

Q2. ***¿Qué indican los offsets 36, 37, 38, 39, 43 y 54 (sugerencia: ver la tabla "Formato del sector de arranque")?. Representar en una tabla similar los valores que toman en la imagen de nuestro disquete.***

OFFSET	36	1 BYTE	Nº de unidad física	00 (0)
OFFSET	37	1 BYTE	Reservado	00 (0)
OFFSET	38	1 BYTE	Valor 29h	29 (41)
OFFSET	39	2 Palabras	Nº de serie de disco	1bed374d (468531021
OFFSET	43	11 BYTES	Título del disco	TORTURATOR
OFFSET	54	8 BYTES	Sistema de ficheros	FAT12

Q3. ***¿Cuál es el tamaño total del disco (en Kilobytes)?. ¿Dónde termina el boot record?.***

· Tamaño del disco:

512 bytes por sector X 2880 sectores = 1474560 bytes totales / 1024 = 1440 Kbytes

· El Boot Record termina en el offset 511, pues es el primer sector y cada sector está formado por 512 bytes.

Q4. ***Analizando la FAT:***

¿Qué tipo de FAT utiliza el disco (12, 16, etc.)?

· FAT12.

¿En qué direcciones empiezan y terminan cada una de las FAT's?

· La primera FAT comienza en el offset 512 y termina en el 5119.

· La segunda FAT, que es la copia de la primera, empieza en el offset 5120 y termina en el 9727.

¿Cuántos ficheros y/o directorios hay en el disco?

· Dos.

¿Cuántos sectores consume cada FAT?, ¿y en kilobytes?

· Según la tabla, cada FAT ocupa 9 sectores.

Es decir, cada una ocupa 4'5 Kbytes = (9 sectores · 512 bytes por sector)/1024.

¿Cuál es el numero máximo de entradas (directorios y/o ficheros) que puede haber en el disquete?

· 224.

¿Existe algún cluster defectuoso?

· No.

Q5. ***Analizando la zona de directorios:***

¿En qué dirección empieza y termina la zona de directorio?

· Comienza en el offset 9728 y termina en el offset 16895.

¿Cuántos sectores ocupa?

· (16895−9728)/512 = 14

tamaño en bytes/512 bytes por sector = 14 sectores

¿Cuántas entradas hay en el directorio raíz? ¿de qué tipo (fichero o directorio)?.

· En el directorio raíz existen 2 entradas:

− la etiqueta de volumen "TORTURATOR"

offset 11 (byte de atributos) = 28

28 = 0010 1000

bit 5: activo si esa entrada de directorio es etiqueta de volumen.

− el directorio "Prac14".

offset 11 (byte de atributos) = 10

10 = 0001 0000

bit 4: activo si es un subdirectorio.

¿Coincide con Q4.3?, ¿por qué?

· No, porque aquí sólo es en el directorio raíz y se cuenta la etiqueta de volumen.

Q6. Finalmente, analizando la zona de datos:

¿Cuál es el contenido de los ficheros y/o directorios?.

· Los ficheros son de texto (extensión txt), y los ficheros "." y ".." son las entradas especiales de cada subdirectorio, en este caso el directorio prac14.

· El fichero ".", que referencia al propio subdirectorio, para que así pueda autolocalizarse.

· El fichero ".." que referencia al directorio padre (cuelga de él).

· El fichero "*host.txt" está borrado, y su contenido es: "Soy recuperable?".

· El fichero "leeme.txt", cuyo contenido es: "Practica 14: Anatomía de un diskette....!"

¿Cuántos ficheros borrados hay?. ¿Es posible determinar exactamente su nombre?, ¿y si empezara por "g"?.

· El único fichero borrado es "*host.txt".

· No podemos determinar su nombre completo porque el primer byte de su nombre se ha machacado con una señal de borrado (e5), por lo que su primera letra la desconocemos. Sólo podemos decir que es ·host.txt.

· Si la primera letra fuera la g, ya podemos decir que el fichero se llama "ghost.txt", reemplazando la señal de borrado e5 por el carácter g en hex.

¿Es posible recuperar el fichero íntegramente?.

· Sí, pues la información del fichero no se ha borrado realmente, sino que se ha indicado que el fichero ha sido borrado en la FAT.