SED:

Ejercicio 1

Con la g son todas las ocurrencias, sin la g solo la primera ocurrencia de cada línea.

cat prueba.txt | sed 's/a/A/g'

Ejercicio 2

cat 2.txt | sed '2d'

Ejercicio 3

a)

cat prueba.txt | sed '3 s/[0-9]//g'

b)

cat 2.txt | sed '2,4 d'

cat 2.txt | sed '3,$ d'

c)

cat 2.txt | sed '3 s/a/A/g'

cat 2.txt | sed '2,$ d'

Ejercicio 4

cat 2.txt | sed 's/.*/& &/g'

SISTEMA DE FICHEROS:

Ejercicio 1

a)

cd

b)
echo $PWD → /home/cursoasr
pwd → /home/cursoasr

c)

cd /usr/bin ; pwd

d)

cd -

d)

cd ../../usr/bin

e)

cd ~

cd *$HOME*

LISTAR CONTENIDOS:

Ejercicio 1

ls -a → Muestra los ficheros ocultos.
ls -d → Lista las características de un directorio pero no de sus contenidos.
ls -l → Long format. Permisos para usuario, grupo y otros. Propietario y fecha.
ls -f → Imprime el contenido del directorio sin ordenar.
ls -F → Da formato a los ficheros listados.
ls -h → Human. Muestra el peso del archivo en KB's
ls -i → i-Node del fichero.
ls -1 → Imprime en una sola columna.
ls - -color → Colorea la salida en función del tipo de fichero.

Ejercicio 2

ls /usr
ls /etc

Ejercicio 3

NOTA: ¿¿¿ Borrar los encabezados ???

ls -i /etc | sort -n > texto4

CREAR Y BORRAR DIRECTORIOS:

Ejercicio 1

mkdir 'mis_archivos' ; ls

Ejercicio 2

mkdir -p mis_archivos/prueba/texto/tmp

Ejercicio 3

rmdir -p mis_archivos/prueba/texto/tmp

COPIAR, BORRAR Y MOVER FICHEROS:

Ejercicio 1

cp texto1 copia1

Ejercicio 2

mkdir copia

mv texto1 texto2 copia1 copia

Ejercicio 3

cp -r copia otra_copia

Ejercicio 4

a)

cd copia ; mv texto1 fichero1

b)

mv fichero1 $HOME ; ls $HOME

c)

mkdir test

d)

mv test /home/cursoasr

e)

cd .. ; rmdir copia

f)

mv fichero1 test/texto1

g)

mv test copia

Ejercicio 5

rm copia/* ; rmdir copia

Ejercicio 6

cd $HOME ; mkdir prueba

mv texto* prueba

cp -r prueba otra_prueba

rm -r prueba otra_prueba ; ls

CARACTERES COMODÍN:

Ejercicio 1

a)

ls texto*

b)

ls -d [0-9]

c)

ls -d *[!34]

d)

cp texto* texto5

ls -d *[!34]

BUSCAR FICHEROS:

Ejercicio 1

find /home -name texto*

Ejercicio 2

find $HOME -type d

Ejercicio 3

find $HOME -size +10M

Ejercicio 4

find $HOME -mtime 0

Ejercicio 5

find $HOME -type f -exec ls -i {} \;

REDIRECCIONES TUBERÍAS Y EXPRESIONES REGULARES:

Ejercicio 1

ls -l text* nada* > salida

El descriptor 0 de ls se reasigna al fichero salida en lugar de al monitor.

Ejercicio 2

cat 1.txt 2.txt > salida.out 2> error.out

Ejercicio 3

cat 1.txt 2.txt >> salida.out 2>> error.out

Ejercicio 4

Diferencias entre “comando > salida 2>&1” y “comando 2>&1 > salida”

El segundo comando es correcto por que se indica correctamente que el descriptor de archivo de stderr y stdout se redirigen al fichero salida.

Ejercicio 5

cat 1.txt 2.txt >> salida.out 2>> /dev/null

Ejercicio 6

Comparar “cat texto1 | sort” con “sort < texto1”

Ambos hacen lo mismo sin embargo:

- En el segundo caso cambia el descriptor de archivo de sort para la entrada.
- En el primer caso se redirecciona la salida del primer comando y la entrada del segundo comando.

Ejercicio 7

```
cat << EOF > ejemplo.txt
> asdfghj
> zxcvbnm
> ;:_
> EOF
```

EXPRESIONES REGULARES:

Ejercicio 1

a)

```
cat texto1 | grep "ja"
```

b)

```
cat texto1 | grep .*ja$
```

c)

```
cat texto1 | grep a.a
```

d)

```
cat texto1 | grep .*al*o.*
```

EL EDITOR VI:

BASH SCRIPTING:

Ejercicio 1

```
#!/bin/bash

if [ $# -eq 2 ]
then
        echo "Nombre del programa: $0"
        echo "Argumentos 1 y 2: $1, $2"
else
        exit 1
fi
exit 0
```

Ejercicio 2

```
#!/bin/bash

if [ $# -eq 1 ] && [ -f $1 ]
then

	NOMBRE=`wc -l $1 | cut -f 2 -d " "`
	LINEAS=`wc -l $1 | cut -f 1 -d " "`
	echo "El fichero $NOMBRE tiene: $LINEAS lineas."

else

	echo "Error, el fichero no es un fichero regular, no existe o el numero de argumentos se ha sobrepasado."

	exit 1

fi

exit 0
```

Ejercicio 3

```
#!/bin/bash

if [ $# -eq 1 ] && [ -d $1 ]
then

	for i in `find $1 -type f` ; do

		echo "Fichero econtrado: $i"

	done

else

	echo "Demasiados propositos o no es un directorio"

	exit 1

fi

exit 0
```

Ejercicio 4

```
#!/bin/bash

function hola(){
        echo “Hola $1!”
}

hola mundo

A=`hola mundo`

echo "La salida es: $A"
```

PROGRAM OUTPUT:

“Hola mundo!”
La salida es: “Hola mundo!”

PROYECTO: AGENDA EN BASH SCRIPT

```
#!/bin/bash

function pause() {

        local NOTHING=""

        echo "Presione una tecla para continuar . . ."

        read NOTHING

        clear

}

function checkname() {

        local REPEATED=0

        for i in `cat agendasr.data | grep $1 | cut -f 1 -d ":"` ; do

                if [ "$i" == "$NOMBRE" ] ; then

                        REPEATED=1
```

```
                break
            fi
        done
        return $REPEATED
}
function checkphone() {
        local DIGITS=0
        local PHONEOK=0
        DIGITS=`echo "$1" | wc -m`
        if [ $DIGITS -lt 10 ] || [ $DIGITS -gt 10 ] ; then
            PHONEOK=1
        fi
        return $PHONEOK
}
function checkmail() {
        local EMAILOK=0
        if [ `echo $1 | tr [@.] [' " '] | wc -w` -ne 3 ] ; then
            EMAILOK=1
        fi
        return $EMAILOK
}
function insert() {
```

```
	local NOMBRE=""
	local CORREO=""
	local TELEFONO=""
	clear
	echo -n "Introduce el nombre: "
	read NOMBRE
	checkname $NOMBRE
	if [ $? -eq 0 ] ; then
		echo -n "Introduce el telefono: "
		read TELEFONO
		checkphone $TELEFONO
		if [ $? -eq 0 ] ; then
			echo -n "Introduce el correo: "
			read CORREO
			checkmail $CORREO
			if [ $? -eq 0 ] ; then
				clear
				echo "$NOMBRE:$TELEFONO:$CORREO" >>
agendasr.data
				echo "El usuario se introdujo correctamente."
			else
				echo "Error, el formato debe ser
<name>@<service>.<domain>"
			fi
```

```
		else
			echo "Error de formato, el contacto debe estar formado por nueve caracteres."
		fi
	else
		echo "El usuario ya existe, no se pudo insertar."
	fi
	pause
}
function delete() {
	local NOMBRE=""
	local RECORD=""
	local RNAME=""
	local LINEA=""
	local DATA=""
	clear
	echo -n "Introduce el nombre por el que borrar: "
	read NOMBRE
	clear
	RECORD=`cat -n agendasr.data | grep "$NOMBRE" | head -n 1`
	RNAME=`echo "$RECORD" | cut -f 2 | cut -f 1 -d ":"`
	if [ "$NOMBRE" != "$RNAME" ]
	then
```

```
		echo "No se encontro ninguna coincidencia."
	else
		LINEA=`echo "$RECORD" | cut -f 1 | sed "s/ //g"`
		DATA=`sed "${LINEA}d" agendasr.data`
		echo "$DATA" > agendasr.data
		echo "El usuario se elimino correctamente."
	fi
	pause
}
function list() {
	clear
	cat agendasr.data
	pause
}
function look() {
	local NOMBRE=""
	local CORREO=""
	local TELEFONO=""
	local RECORD=""
	clear
	echo -n "Introduce el correo por el que buscar: "
	read CORREO
	clear
```

```
        RECORD=`cat agendasr.data | grep $CORREO | head -n 1`

        if [ "$RECORD" == "" ] ; then

                echo "No se encontro ninguna coincidencia."

        else

                NOMBRE=`echo $RECORD | cut -f 1 -d ":"`
                CORREO=`echo $RECORD | cut -f 3 -d ":"`
                TELEFONO=`echo $RECORD | cut -f 2 -d ":"`

                echo "Nombre: $NOMBRE."
                echo "Telefono: $TELEFONO."
                echo "Correo: $CORREO."

        fi

        pause

}

function count() {

        clear

        local COUNT=0

        COUNT=`wc -l agendasr.data | cut -f 1 -d " "`

        echo "Hay un total de $COUNT usuarios."

        pause

}

function menu() {

        local OPTIONS=""

        OPTIONS="Insertar Eliminar Listar Buscar Contar Salir"

        select opt in $OPTIONS ; do
```

```
        case $opt in

            "Insertar") insert ;;
            "Eliminar") delete ;;
            "Listar") list ;;
            "Buscar") look ;;
            "Contar") count ;;
            "Salir") exit 0 ;;

        esac

    done

}

function main() {

    clear

    menu

}

main
```