

EJERCICIOS 2 - LINUX

21.- Crea dos ficheros llamados **and** y **or** en el directorio **HARD** creado en la pregunta 1 de los ejercicios y compáralos. ¿Qué resultado se obtiene cuando los dos ficheros son iguales? En los ficheros and y or escribiremos una breve descripción del funcionamiento de las puertas lógicas and y or.

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ diff and.txt or.txt
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ administrador@U2004:~/Javi
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ echo "la salida es 1 cuando
o ambas salidas son 1" > and.txt
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ echo "al menos una de las
entradas ha de ser 1" > or.txt
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ diff and.txt or.txt
1c1
< la salida es 1 cuando ambas salidas son 1
---
> al menos una de las entradas ha de ser 1
```

Los he creado primero con la palabra hola dentro para ver que sucede al compararlos siendo iguales, y luego he metido dentro una breve descripción de las puertas lógicas y los he vuelto a comparar con el diff, para ver que sucedía.

22.- Mediante el comando diff visualizar cuál es la diferencia entre los ficheros **and** y **or**. Posteriormente realizar la captura con el comando sdiff.

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ sdiff and.txt or.txt
la salida es 1 cuando ambas salidas son 1          | al menos una de
las entradas ha de ser 1
```

Con el sdiff la diferencia es esa

23.- Mediante el comando man deseamos conocer deseamos información de los comandos siguientes: mkdir, Sed, vi, nano, cat.

```
MKDIR(1)                                User Commands                                MKDIR(1)

NAME
    mkdir - make directories

SYNOPSIS
    mkdir [OPTION]... DIRECTORY...

DESCRIPTION
    Create the DIRECTORY(ies), if they do not already exist.

    Mandatory arguments to long options are mandatory for short options
    too.

    -m, --mode=MODE
        set file mode (as in chmod), not a=rwx - umask

    -p, --parents
        no error if existing, make parent directories as needed

    -v, --verbose
        print a message for each created directory

Manual page mkdir(1) line 1 (press h for help or q to quit)
```

Esta es la ayuda que se muestra al poner man mkdir.

EJERCICIOS 2 - LINUX

```
SED(1)                                User Commands                                SED(1)

NAME
    sed - stream editor for filtering and transforming text

SYNOPSIS
    sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION
    Sed is a stream editor. A stream editor is used to perform basic text
    transformations on an input stream (a file or input from a pipeline).
    While in some ways similar to an editor which permits scripted edits
    (such as ed), sed works by making only one pass over the input(s), and
    is consequently more efficient. But it is sed's ability to filter text
    in a pipeline which particularly distinguishes it from other types of
    editors.

    -n, --quiet, --silent

        suppress automatic printing of pattern space

    --debug

Manual page sed(1) line 1 (press h for help or q to quit)
```

Esta es la ayuda que se muestra utilizando el `man` con el comando `sed`

```
VIM(1)                                General Commands Manual                                VIM(1)

NAME
    vim - Vi IMproved, a programmer's text editor

SYNOPSIS
    vim [options] [file ...]
    vim [options] -
    vim [options] -t tag
    vim [options] -q [errorfile]

    ex
    view
    gvim gview evim eview
    rvim rview rgvim rgview

DESCRIPTION
    Vim is a text editor that is upwards compatible to Vi. It can be used
    to edit all kinds of plain text. It is especially useful for editing
    programs.

    There are a lot of enhancements above Vi: multi level undo, multi win-
    dows and buffers, syntax highlighting, command line editing, filename

Manual page vi(1) line 1 (press h for help or q to quit)
```

Esta es la ayuda que se muestra con el comando `man` al comando `vi`

EJERCICIOS 2 - LINUX

```
NANO(1)                                General Commands Manual                                NANO(1)

NAME
    nano - Nano's ANOther editor, inspired by Pico

SYNOPSIS
    nano [options] [[+line[,column]] file]...

    nano [options] [[+[crCR](/|?)string] file]...

NOTICE
    Since version 4.0, nano by default:

    • does not automatically hard-wrap lines that become overlong,
    • includes the line below the title bar in the editing area,
    • does linewise (smooth) scrolling.

    If you want the old, Pico behavior back, you can use --breaklonglines,
    --emptyline, and --jumpyscrolling (or -bej for short).

DESCRIPTION
    nano is a small and friendly editor. It copies the look and feel of
    Pico, but is free software, and implements several features that Pico
    Manual page nano(1) line 1 (press h for help or q to quit)
```

Esta es la ayuda del comando nano

```
CAT(1)                                User Commands                                CAT(1)

NAME
    cat - concatenate files and print on the standard output

SYNOPSIS
    cat [OPTION]... [FILE]...

DESCRIPTION
    Concatenate FILE(s) to standard output.

    With no FILE, or when FILE is -, read standard input.

    -A, --show-all
        equivalent to -vET

    -b, --number-nonblank
        number nonempty output lines, overrides -n

    -e
        equivalent to -vE

    -E, --show-ends
        display $ at end of each line
    Manual page cat(1) line 1 (press h for help or q to quit)
```

Esta es la ayuda del comando cat

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man mkdir
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man sed
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man vi
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man nano
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man cat
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$
```

Estos son todas las capturas del man

EJERCICIOS 2 - LINUX

24.- Deseamos buscar todos aquellos manuales donde se encuentre la palabra awk.

```
MAWK(1)                                USER COMMANDS                                MAWK(1)

NAME
    mawk - pattern scanning and text processing language

SYNOPSIS
    mawk [-W option] [-F value] [-v var=value] [--] 'program text' [file
    ...]
    mawk [-W option] [-F value] [-v var=value] [-f program-file] [--] [file
    ...]

DESCRIPTION
    mawk is an interpreter for the AWK Programming Language. The AWK lan-
    guage is useful for manipulation of data files, text retrieval and pro-
    cessing, and for prototyping and experimenting with algorithms. mawk
    is a new awk meaning it implements the AWK language as defined in Aho,
    Kernighan and Weinberger, The AWK Programming Language, Addison-Wesley
    Publishing, 1988 (hereafter referred to as the AWK book.) mawk con-
    forms to the POSIX 1003.2 (draft 11.3) definition of the AWK language
    which contains a few features not described in the AWK book, and mawk
    provides a small number of extensions.

    An AWK program is a sequence of pattern {action} pairs and function

Manual page awk(1) line 1 (press h for help or q to quit)
```

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ apropos awk
awk (1)          - pattern scanning and text processing language
mawk (1)         - pattern scanning and text processing language
nawk (1)         - pattern scanning and text processing language
```

```
MAWK(1)                                USER COMMANDS                                MAWK(1)

NAME
    mawk - pattern scanning and text processing language

SYNOPSIS
    mawk [-W option] [-F value] [-v var=value] [--] 'program text' [file
    ...]
    mawk [-W option] [-F value] [-v var=value] [-f program-file] [--] [file
    ...]

DESCRIPTION
    mawk is an interpreter for the AWK Programming Language. The AWK lan-
    guage is useful for manipulation of data files, text retrieval and pro-
    cessing, and for prototyping and experimenting with algorithms. mawk
    is a new awk meaning it implements the AWK language as defined in Aho,
    Kernighan and Weinberger, The AWK Programming Language, Addison-Wesley
    Publishing, 1988 (hereafter referred to as the AWK book.) mawk con-
    forms to the POSIX 1003.2 (draft 11.3) definition of the AWK language
    which contains a few features not described in the AWK book, and mawk
    provides a small number of extensions.

-----Info: (*manpages*)awk, 1081 lines --Top-----
No existe ningún elemento de menú llamado 'awk' en el nodo '(dir)Top'
```

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man awk
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ apropos awk
awk (1)          - pattern scanning and text processing language
mawk (1)         - pattern scanning and text processing language
nawk (1)         - pattern scanning and text processing language
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ help awk
bash: help: no hay temas de ayuda que coincidan con 'awk'. Pruebe `help help' o
`man -k awk' o `info awk'.
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ info awk
```

Estas son todas las ayudas utilizadas con su respectiva pantalla

EJERCICIOS 2 - LINUX

25.- Utilizando el comando `help` deseamos mostrar una breve información de los comando: `mkdir`, `while`, `who`, `man` y `apropos`. ¿Qué diferencia existe con el comando `man`?

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ help mkdir
bash: help: no hay temas de ayuda que coincidan con `mkdir'. Pruebe `help help'
o `man -k mkdir' o `info mkdir'.
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ help while
while: while ÓRDENES; do ÓRDENES; done
Ejecuta órdenes mientras una prueba sea correcta.

Expande y ejecuta ÓRDENES mientras la orden final en las ÓRDENES
`while' tenga un estado de salida de cero.

Estado de Salida:
Devuelve el estado de la última orden ejecutada.
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ help who
bash: help: no hay temas de ayuda que coincidan con `who'. Pruebe `help help' o
`man -k who' o `info who'.
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ help man
bash: help: no hay temas de ayuda que coincidan con `man'. Pruebe `help help' o
`man -k man' o `info man'.
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ help apropos
bash: help: no hay temas de ayuda que coincidan con `apropos'. Pruebe `help hel
p' o `man -k apropos' o `info apropos'.
```

La diferencia es que la ayuda es muy breve y no te la muestra en casi ningún comando

26.- Mediante el comando **apropos** , deseamos obtener información de aquellos comandos que se relacionan con el comando **grep**

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ apropos grep
bzegrep (1) - search possibly bzip2 compressed files for a regular e...
bzfgrep (1) - search possibly bzip2 compressed files for a regular e...
bzgrep (1) - search possibly bzip2 compressed files for a regular e...
egrep (1) - print lines that match patterns
fgrep (1) - print lines that match patterns
grep (1) - print lines that match patterns
lzegrep (1) - search compressed files for a regular expression
lzfgrep (1) - search compressed files for a regular expression
lzgrep (1) - search compressed files for a regular expression
pgrep (1) - look up or signal processes based on name and other at...
ptargrep (1) - Apply pattern matching to the contents of files in a t...
rgrep (1) - print lines that match patterns
xzegrep (1) - search compressed files for a regular expression
xzfgrep (1) - search compressed files for a regular expression
xzgrep (1) - search compressed files for a regular expression
zegrep (1) - search possibly compressed files for a regular expression
zfgrep (1) - search possibly compressed files for a regular expression
zgrep (1) - search possibly compressed files for a regular expression
zipgrep (1) - search files in a ZIP archive for lines matching a pat...
```

Esto es todo lo que aparece al poner `apropos grep`

EJERCICIOS 2 - LINUX

27.- ¿Qué información conseguiríamos si escribimos el comando siguiente:

\$ *apropos* calculator

¿Qué diferencia existe con el comando `man -k`?

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ apropos calculator
bc (1) - An arbitrary precision calculator language
dc (1) - an arbitrary precision calculator
mate-calc (1) - (mate-calculator) - The MATE Desktop Environment Calcu...
mate-calc-cmd (1) - A console calculator for the MATE Desktop Environment.
mate-calculator (1) - (mate-calculator) - The MATE Desktop Environment Calcu...
transicc (1) - little cms ColorSpace conversion calculator.
xcalc (1) - scientific calculator for X
```

Esto aparece al poner `apropos calculator`

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/HARD$ man -k calculator
bc (1) - An arbitrary precision calculator language
dc (1) - an arbitrary precision calculator
mate-calc (1) - (mate-calculator) - The MATE Desktop Environment Calcu...
mate-calc-cmd (1) - A console calculator for the MATE Desktop Environment.
mate-calculator (1) - (mate-calculator) - The MATE Desktop Environment Calcu...
transicc (1) - little cms ColorSpace conversion calculator.
xcalc (1) - scientific calculator for X
```

Esto aparece al poner `man -k calculator`, exactamente lo mismo; no hay diferencia alguna.

28.- Mediante el comando `cat` crear un fichero llamado **ftp** en el directorio **Sistemas** y escribir una pequeña explicación de la función de lo que hace el servidor **ftp**. (el fichero debe contener al menos 4 líneas)

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1$ cd SISTEMA/
administrador@U2004:~/Javier_Arruabarrena/ASIR1/SISTEMA$ cat > ftp.txt
Las siglas FTP significan "File Transfer Protocol", que se traduciría como "Prot
ocolo de Transferencia de Archivos". El servicio FTP es un servicio utilizado pa
ra el envío y obtención de archivos entre dos equipos remotos.
^C
```

*He hecho un `cd` a `sistema` para crear el archivo ya en dicha carpeta con el comando `cat` y escribiendo una breve descripción del servidor **FTP**.*

EJERCICIOS 2 - LINUX

29.- Mediante el comando **nano** editar el fichero **ftp** creado en el ejercicio anterior.



```
GNU nano 4.8 ftp.txt Modificado
Las siglas FTP significan "File Transfer Protocol", que se traduciría como "Pro
Estoy creando más texto en el archivo.

Ver ayuda  Guardar  Buscar  Cortar Tex  Justificar  Posición
Salir  Leer fich.  Reemplazar  Pegar  Ortografía  Ir a línea
```

He metido más texto con el comando nano, puede verse en el texto y en la parte superior derecha que pone modificado. He hecho ctrl+O para guardar y ctrl+X para salir del nano.

30.- Explicar qué hace la siguiente orden:

```
# cat /home/texto.txt /home/noexisto.txt 2>/home/error.txt >
/home/concatena.txt
```

Está concatenando todo el contenido de los archivos texto.txt, noexisto.txt y error.txt en concatena.txt

31.- Instalar la aplicación **gedit** en caso de no tenerlo instalado y editar el fichero creado en el ejercicio 28.

Lo he instalado poniendo sudo apt-get install gedit y poniendo la contraseña del administrador. Después he puesto gedit ftp.txt y me aparece esto:



```
Abrir + ftp.txt Guardar
~(javier_Arruabarrena/ASIR1/SISTEMA)
1 Las siglas FTP significan "File Transfer Protocol", que se traduciría como "Protocolo de
Transferencia de Archivos". El servicio FTP es un servicio utilizado para el envío y obtención
de archivos entre dos equipos remotos.
2 Estoy creando más texto en el archivo.

Texto plano  Anchura del tabulador: 8  Ln 1, Col 1  INS
```

EJERCICIOS 2 - LINUX

```
1 Las siglas FTP significan "File Transfer Protocol", que se traduciría como "Protocolo de
  Transferencia de Archivos". El servicio FTP es un servicio utilizado para el envío y obtención
  de archivos entre dos equipos remotos.
2 Estoy creando más texto en el archivo.
3 Vuelvo a editar el archivo
```

Lo he editado con el gedit poniendo una oración nueva.

32.- Existen otros editores como (**nano**, **vi**, **Sed**). Comprueba cuál es el resultado con estos editores.

```
GNU nano 4.8      ftp.txt
Las siglas FTP significan "File Transfer Protocol", que se traduciría como "Pro
Estoy creando más texto en el archivo.
Vuelvo a editar el archivo

[ 3 líneas leídas ]
^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar Tex^J Justificar^C Posición
^X Salir      ^R Leer fich.^_ Reemplazar^U Pegar      ^T Ortografía^_ Ir a línea
```

Así aparece con el nano

[illegible]

Así aparece con el vi

EJERCICIOS 2 - LINUX

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/SISTEMA$ sed -e ftp.txt
sed: -e expresión #1, carácter 1: orden desconocida: «f»
```

Con el comando `sed` da esta especie de error.

33.- Deseamos contar el número de líneas del fichero **ftp** creado en el ejercicio 28, así mismo contar el número de palabras y por último el número de caracteres.

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/SISTEMA$ wc ftp.txt
 3  46 294 ftp.txt
```

Poniendo `wc` y el nombre del archivo ya lo hace, sin tener que poner ningún parámetro

34.- Instalar el comando **tree** en caso de no tenerlo instalado y mostrar solo los directorios de ASIR1

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/SISTEMA$ tree ..
..
├── BASEDATOS
├── FOL
├── HARD
│   ├── and.txt
│   ├── BOOLE
│   ├── HOFMANN
│   └── or.txt
├── REDES
└── SISTEMA
    ├── ftp.txt
    ├── IOS
    └── LINUX

9 directories, 3 files
```

Ya está instalado así que he hecho un `tree` con dos puntos para ir a la carpeta anterior

35.- ¿Qué diferencia existe entre **tree** y **pstree**?

Que `tree` te muestra el árbol de directorios y `pstree` te muestra el árbol de procesos

36.- Mostrar el mes del año de tu nacimiento

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/SISTEMA$ cal -m 05 2003
    Mayo 2003
do lu ma mi ju vi sa
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

He utilizado el comando `cal` con el atributo `-m` y el mes y año de mi nacimiento para que me mostrase dicho mes.

EJERCICIOS 2 - LINUX

37.- Nos encontramos en el mes de Abril, deseamos que se nos muestre el mes actual, así como el mes anterior y posterior.

```
administrador@U2004:~/Javier_Arruabarrena$ cal -m 4 -3
      Marzo 2022      Abril 2022      Mayo 2022
do lu ma mi ju vi sá do lu ma mi ju vi sá do lu ma mi ju vi sá
      1  2  3  4  5      1  2      1  2  3  4  5  6  7
 6  7  8  9 10 11 12  3  4  5  6  7  8  9  8  9 10 11 12 13 14
13 14 15 16 17 18 19 10 11 12 13 14 15 16 15 16 17 18 19 20 21
20 21 22 23 24 25 26 17 18 19 20 21 22 23 22 23 24 25 26 27 28
27 28 29 30 31      24 25 26 27 28 29 30 29 30 31
```

38.- Utilizando el comando `date` hacer captura del resultado conseguido y posteriormente deseamos construir mediante el mismo comando la fecha actual con el resultado:

Monday 13 April

```
administrador@U2004:~/Javier_Arruabarrena/ASIR1/SISTEMA$ date
mié 21 sep 2022 14:34:14 CEST
```

```
inf2@Salesianas:~$ date +"%A %d %B"
miércoles 28 septiembre
```

39.- Deseamos conseguir el mensaje que nos muestre la siguiente información:

“Estamos en Madrid a % **día** de %**mes** de %**año** y la hora es %**hora**”

```
administrador@U2004:~/Javier_Arruabarrena$ date +"Estamos en Madrid a %A de %d de %y y la hora es %R"
Estamos en Madrid a miércoles de 28 de 22 y la hora es 13:58
```

40.- Mediante el comando **grep** muestra las líneas de un archivo que contenga una palabra del archivo `and` que hayas utilizado para describir el funcionamiento de una puerta `or` del ejercicio

```
administrador@U2004:~/Javier_Arruabarrena$ grep ser ./ASIR1/HARD/or.txt
al menos una de las entradas ha de ser 1
```