IES Rodrigo Caro

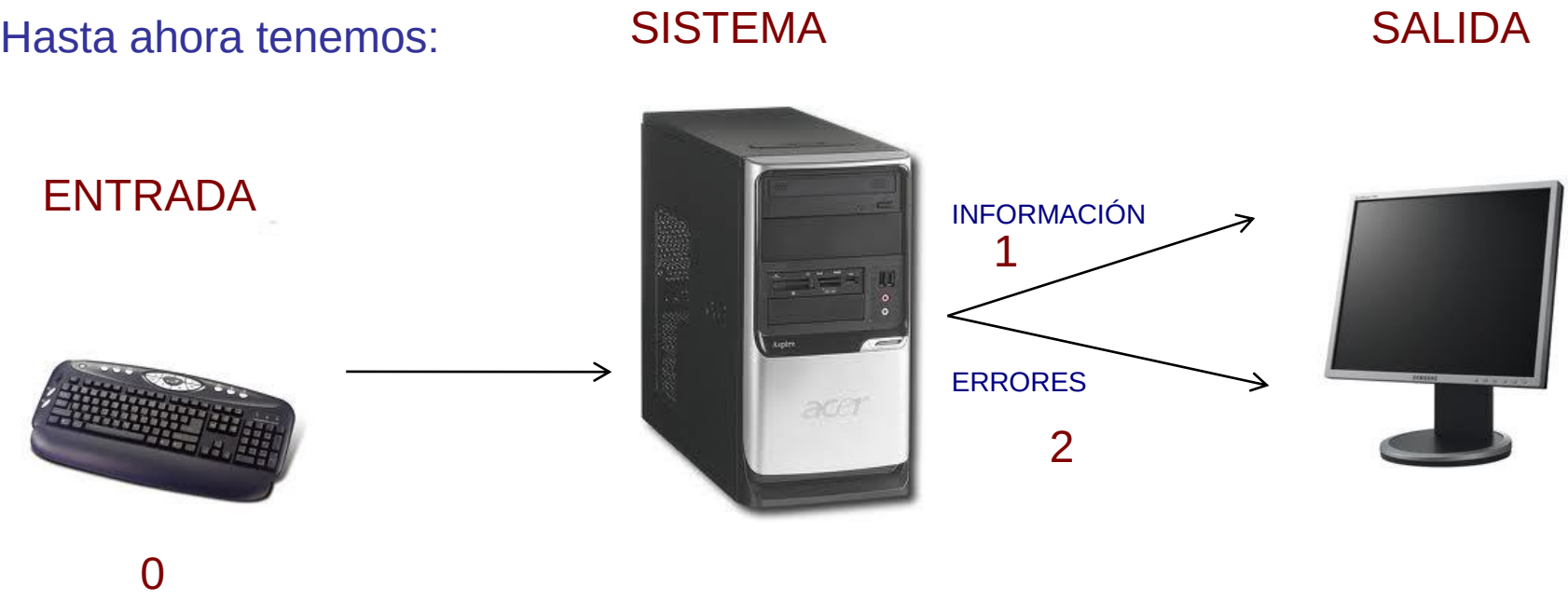
Dpto de Informática

ud9.- Redireccionamientos y filtros.

Índice:

1.- Introducción.

Hasta ahora tenemos:



La información a procesar o las órdenes las introducíamos por teclado y el resultado nos lo mostraba por pantalla.

Ahora vamos a ver cómo la entrada y la salida se puede redireccionar.

También veremos cómo extraer información de los archivos con la ayuda de filtros.

2.- Redireccionamientos.

2.1.- Redireccionar la salida:

$ comando > archivo | dispositivo

$ comando >> archivo

2.2.- Redireccionar los errores:

$ comando 2> archivo | dispositivo

$ comando 2>> archivo

2.3.- Redireccionar la entrada:

$ comando < información

Símbolo	Descripción	Sintaxis
<	Redirección de entrada	Orden < fichero_ dispositivo
>	Redirección de salida	Orden > fichero_ dispositivo
>>	Adición	Orden >> fichero_ dispositivo

>>

Si el fichero no existe lo crea.
Si el fichero existe añade la información al final.

/dev/null

es un archivo vacío que se utiliza para recoger mensajes que no necesitamos que aparezca en pantalla.

2.- Redireccionamientos.

Ejercicios:

1.- Abre varias sesiones en el equipo: heidi, pedro y usuario.

2.- Necesitamos generar un informe en el que aparezca la fecha del sistema y los usuarios que en ese momento se encuentran conectados.

3.- El informe se llamará informe.txt

4.- Realizar un pequeño script que realice ese proceso.

3.- Concatenar y filtrar.

La concatenación nos permite que la salida de un comando sea interpretada por otro comando.

El operador que utilizaremos para ello es:

Operador tuberia | (Alt Gr+1)

3.1.-More y less

Nos muestra la información paginada.

$ comando | more

$ comando |less

more

Barra espaciadora:Av Pág
Enter: avanza una línea
Q:salir

less

Av Pág, Re Pág, ↓↑
/cadena → Busca una cadena
Nos permite acceder a la información mostrada en la pantalla anterior.

3.- Concatenar y filtrar.

3.2.- pr

Sirve para darle formato a un archivo.

$ pr [opciones] fichero $comando |pr [opciones]

Opciones:

-h “texto”	→ Poner un título .Ejemplo: -h “ “ Línea en blanco.
-d	→ Doble espacio.
-w nº	→ Ancho de la página en caracteres.
-l nº	→ Longitud de la página en líneas.
-n	→ líneas enumeradas.

3.- Concatenar y filtrar.

3.3.- sort

Nos permite ordenar el contenido de uno o varios ficheros.

$ sort [opciones] fichero $comando |sort [opciones]

Opciones:

- -n → ordenación numérica.
- -u → Elimina líneas repetidas.

3.- Concatenar y filtrar.

3.3.- sort

3.3.1.- sort con campos

$ sort –t □ -k □ fichero

Separador de campos

Nº del campo a ordenar

Cuando el separador de campos es un espacio o un tabulador no hay que indicar nada.

Ejemplos:

$ sort –t, -k2 prueba → ordena por el campo 2

$ sort –t: -k 2,4 prueba → ordena por los campos 2,3,4

$ sort –t: -k2 –k4 prueba → ordena primero por el 2 y luego por 4.

3.- Concatenar y filtrar.

Ejercicios.

3.3.1 sort con campos

1.- Creamos el siguiente fichero llamado alumnos.
Domínguez López, Ángel
Díaz casado, Luisa
Casado Benjumea, Carmen
Diáz caza, Antonio

2.- Ordénalo.

3.- Añadir los siguientes campos, ordenarlos por ciudad.
……Ángel,Sevilla
……Luisa,Málaga
……Carmen,Cádiz
……Antonio,Málaga

4.- Ordenarlos por ciudad y nombre.

5.- Añadir los campos y ordenarlo por edad.
……Sevilla,9
……Málaga,20
……Cádiz,8
…..Málaga,20

6.-Ordenarlo por edad, ciudad y nombre.

3.- Concatenar y filtrar.

Ejercicios.

ACTIVIDAD

1.- Creamos el siguiente archivo llamado telefonos.
Ana:954 22 22 22:Sevilla
Margarita:954 11 11 11:Málaga
Pedro:954 33 33 33:Cádiz
Ana:954 22 22 11:Sevilla

2.-Copiar ese archivo a uno llamado agenda. Vamos hacer pruebas sobre este otro.

3.- ordenar el archivo agenda por nombre, ciudad y telefono y además el resultado volcarlo sobre el propio agenda. Investigar en las opciones del sort.

3.- Concatenar y filtrar.

3.4.- tr

Nos permite cambiar unos caracteres por otros.

Sustituye el conj1 por el conj2

$ tr [opciones] conj1 conj2 $comando |tr [opciones] conj1 conj2

Opciones:

-s → Sustituye un conjunto de caracteres repetidos por uno sólo.

-c → Sustituye todos los caracteres que no se especifiquen en el conj1 (lo contrario a lo que se indique).

-d → Elimina caracteres.

3.- Concatenar y filtrar.

Ejercicios.

1.- Crea el siguiente archivo:

```
usuario@servidor200:~$ cat datos.txt
mmanzano:mmanzano:María:Manzano:mmanzano@gmail.com
mdominguez: mdominguez:    Manuel:  Dominguez:mdominguez@gmail.com
rcarmona:     rcarmona: Rafael:Carmona:   rccarmona@gmail.com
lperez:  lperez:Luisa:Perez:   lperez@gmail.com
```

2.- El resultado debe ser:

```
mmanzano:mmanzano:María:Manzano:mmanzano@gmail.com
mdominguez:mdominguez:Manuel:Dominguez:mdominguez@gmail.com
rcarmona:rcarmona:Rafael:Carmona:rccarmona@gmail.com
lperez:lperez:Luisa:Perez:lperez@gmail.com
```

3.5.- sed

Nos permite sustituir una cadena por otra

- Modificar líneas

```
$ sed s/cadena1/cadena2/ archivo_entrada
$ sed s/cadena1/cadena2/g archivo_entrada
```

- Modifica las apariciones de cadena1 por cadena2

- Usando g modifica todas las apariciones de una línea

3.- Concatenar y filtrar.

Ejercicios.

1.- Crea el siguiente archivo, que se trata de la configuración de la red.

```
usuario@servidor200:~$ cat red.txt
Configuración de la red
address 192.168.1.50
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

2.- La configuración de la red ha cambiado y debemos sustituir 192.168.1 por 172.26.0 El resultado debe ser:

```
Configuración de la red
address 172.26.0.50
netmask 255.255.255.0
broadcast 172.26.0.255
gateway 172.26.0.1
usuario@servidor200:~$
```

4.- Búsqueda de información.

4.1.- Opciones básicas del grep

El comando más utilizado es el grep. Se utiliza para buscar coincidencias en una línea.

$grep [opciones] patrón fichero $cat fichero|grep [opciones] patrón

Opciones:

-i → Ignora mayúsculas o minúsculas.

-v → Devuelve las líneas que no contienen el patrón.

-c → Cuenta las líneas en las que se ha encontrado el patrón

-r →¿Para qué puede ser interesante esta opción?

4.- Búsqueda de información.

4.1.- Opciones básicas del grep

$grep [opciones] patrón fichero $cat fichero|grep [opciones] patrón

Patrón:

" " → Utilizaremos comillas para buscar un patrón.

"^ cadena" → Busca las líneas que comienzan por la cadena.

"cadena$" → Busca las líneas que terminan por la cadena.

. → Representa un carácter.

.* → Representa cualquier conjunto de caracteres.

Ejemplo: ¿Qué buscaría este patrón: " .*:.*:.*p:.*"

4.- Búsqueda de información.

Ejercicios.

4.- El comando grep

1.- Escribimos el siguiente texto y lo guardamos en el fichero comerciales.

García Pérez:María:Valencia
Martín García:Sandra:Madrid
Romero Redondo:Carlos:Barcelona
Martín garcía:Pedro:madrid
Soriano Frías:Manuel:Barcelona
Domínguez Martín:Ana:Madrid

2.- Averiguar los comerciales que trabajan en Barcelona.

3.- Comerciales que no trabajen en Madrid.

4.- Comerciales que no trabajen en Madrid ni en Barcelona.

5.- Todos los comerciantes cuyo apellido empiece por García.

6.- Comerciantes de Madrid cuyo 1er apellido no empiece por Martín.

4.- Búsqueda de información.

Ejercicios.

4.- El comando grep

1.- Escribimos el siguiente texto y lo guardamos en el fichero comerciales.

García Pérez:María:Valencia
Martín García:Sandra:Madrid
Romero Redondo:Carlos:Barcelona
Martín garcía:Pedro:madrid
Soriano Frías:Manuel:Barcelona
Domínguez Martín:Ana:Madrid

Imagínate que el registro 3 fuese: Romero Barcelona:Carlos:Barcelona

¿Podríamos sacar los comerciantes que viven en Barcelona?

awk

4.- Búsqueda de información.

Ejercicios.

1.- Visualiza el fichero de configuración del editor nano /etc/nanorc

2.- ¿Cuántas líneas tienes?

3.- ¿Serías capaz de visualizarlo sin mostrar las líneas de comentarios?

Nota: Las líneas de comentarios empiezan por #

4.- Búsqueda de información.

4.2.- Operador or

Operador OR: \| (Barra invertida y operador tubería)

Ejemplo: Extraer las líneas que contenga la cadena agua o bebida

$cat prueba | grep “agua\|bebida”

Utilidad:

Muchos de los archivos de configuración contienen líneas comentadas y líneas en blanco. Podemos hacer que con el comando grep no nos la muestre.

4.- Búsqueda de información.

4.2.- Operador or

Ejemplo:

Vamos a simular que tenemos un fichero de configuración.

1.- Crea el siguiente archivo, llamado configuracion:

```
# fichero de configuración
# Definición de la variable a
Set a=1
# Definición de la variable b
Set b=2


#
Set c=3
```

2.- Observas que hay varias líneas que empiezan por # y dos líneas en blanco.

3.- Utilizando el grep, mostrar, aquellas líneas que no comienzan por # ni líneas blancas.

Nota: Línea en blanco sería equivalente a poner "^$", líneas que comienzan y acaban

5.- Manipulación de ficheros de texto.

Vamos a ver cómo extraer información de los ficheros de texto.

5.1.- cut

Sirve para seleccionar una o varias columnas de un fichero.

$ cut –d □ -f □ fichero $cat fichero | cut –d □ -f □

Separador de campos

Campo a extraer.

Cuando el separador de campos es un tabulador no hay que indicar nada. Si es un espacio debe ir entre ‘ ‘

Ejemplos:

$cat prueba|cut –d: -f1,3,5 → Selecciona los campos 1,3,5,

$cat prueba|cut –d: ‘ ‘ –f3-5 → Selecciona los campos 3,4 y 5

5.- Manipulación de ficheros de texto.

Ejercicios.

5.1.- El comando cut

1.- Borrar el contenido del directorio de trabajo excepto el Desktop y el archivo comerciales.

2.- Visualizar el fichero comerciales y escribir la estructura de campos.

3.- Seleccionar solo los nombres.

4.- Seleccionar los nombres y la ciudad, y además redireccionarlos a un fichero llamado direcciones.

5.- Seleccionar ciudades y apellidos en este orden. ¿Puedes?

5.- Manipulación de ficheros de texto.

5.2.- paste

Nos permite unir distintos ficheros, línea por línea.

$ paste [opciones] fichero1 fichero2

Opciones:

-d □ → es el separador en el fichero destino.

5.- Manipulación de ficheros de texto.

Ejercicios.

5.2.- El comando paste

1.- Crear los siguientes archivos:

nombres	Correo	dpto
MARISA SÁNCHEZ	marsan@terra.es	Ventas
CARLOS GARCÍA	cargar@terra.es	Ventas
PILAR MOLINA	pilmol@terra.es	Compras
VICTORIA GIL	vicgil@terra.es	Almacen
MARIANO MORENO	marmor@terra.es	Marketing

2.- Mostrar la unión de los tres ficheros, utilizando como separador :

3.- Redireccionar la salida a un fichero de salida llamado empleados.

5.- Manipulación de ficheros de texto.

Ejercicios: cut y paste

En el servidor se ha instalado el servicio de directorio OpenLDAP.

Los datos de los usuarios nos lo muestran de la siguiente forma:

usuarios.ldap

uid:alopez
uidNumber:2007
gidNumber:2000
homeDirectory:/home/alopez
uid:mdominguez
uidNumber:2008
gidNumber:2000
homeDirectory:/home/mdominguez

Necesitamos un archivo con el siguiente formato:

```
uid:alopez:uidNumber:2007:gidNumber:2000:homeDirectory:/home/alopez
uid:mdominguez:uidNumber:2008:gidNumber:2000:homeDirectory:/home/mdominguez
```

5.- Manipulación de ficheros de texto.

5.3.- awk

Realmente es un lenguaje de programación pero nosotros lo utilizaremos para extraer información de un fichero en un determinado orden.

$ awk –F □ '[condición] {orden}' fichero

↓

Campo de separación

Operaciones lógicas

&& → Operación Y

|| → Operación OR

Ejemplos:

$awk –F: '{print $5 "-" $4}' fichero → Extrae el campo 5 y el campo 4, separados por - .

$awk –F: '{print $0}' fichero → Extrae todos los campos.

$awk –F: '$1==1000 {print $2}' fichero → Si el campo1 es igual a 1000, extrae el campo 2.

$awk –F: '$1=1000 {print $2}' fichero → Asígnale al campo1 el valor 1000 y después muestra el campo2.

$awk -F: '$5 > 100 && $5 <= 200 {print $8}' fichero → Si el campo5 es mayor que 100 y menor o igual que 200, extrae el campo 8.

$ awk –F: '$2!="ES" {print $1}' fichero → Si el campo 2 es distinto de ES, extrae el campo 1.

5.- Manipulación de ficheros de texto.

Ejercicios.

.

5.3 El comando awk

1.- Visualizamos comerciales y apuntamos su estructura.

2.- Extraer en el orden: ciudad y nombre.

3.- Extraer la misma información, pero ahora separando los campos por un guión.

4.- Añade al fichero comerciales, un campo más que sea la edad.

20, 50, 25, 42, 36, 22

5.- Extraer los nombres de los comerciantes que sean mayores o igual de 30 años.

5.- Manipulación de ficheros de texto.

5.4.- join

Nos permite unir los datos de dos ficheros.

Condiciones para unirlos:

.- Los ficheros deben tener un campo en común.

.- Los ficheros deben estar ordenador por ese campo.

$join –t □ -1□ -2 □ fichero1 fichero2

Delimitador campo

Campo común 1er fichero

Campo común en 2º fichero

5.- Manipulación de ficheros de texto.

Ejercicios.

5.4.- El comando join

1.- Creamos los archivos datos y email.

Datos	Email
MARISA SÁNCHEZ:111A:954 11 11 11:SEVILLA CARLOS GARCÍA:222B:954 22 22 22:CÁDIZ PILAR MOLINA:33C:954 33 33 33:SEVILLA	111A:MARSAN@TERRA.ES 222B:CARGAR01@TERRA.ES 22B:CARGAR02@GMAIL.COM

2.- Ver si se dan las condiciones para unirlos.

3.- Unir ambos ficheros.

4.- Necesito los datos de los que viven en Sevilla.

5.- Necesito el nombre, el teléfono y el correo de los que viven en Sevilla.

5.- Manipulación de ficheros de texto.

Ejercicio:

Acabas de empezar tus prácticas de empresa y te preguntan si manejas Linux, a lo que tú naturalmente respondes que sí ☺.

Te plantean el siguiente trabajo:

Tenemos dos archivos.

Alumnos.txt (NºID:Nombre:Apellidos)

0001:Manuel:Dominguez
0002:Rosa:Perez
0004:Pedro:García
0003:Luis:Millar

notas.txt (NºID:Nota_sistemas:Nota_BD)

0001:6:8
0002:7:10
0004:8:8

Hay que generar un fichero con las notas de sistemas:

Notas de sistemas:

0001:Manuel:Domínguez:6
0002:Rosa:Perez:7
0004:Pedro:García:8