

# Comandos básicos Linux II

## Comandos Básicos II

---

- Filtros:
  - Permite escribir y leer datos de los canales estándares de entrada y salida.
  - **More** muestra los datos página a página.
  - **Sort** permite ordenar los datos
  - **Grep** especifica criterios de búsqueda.

## Comandos Básicos II

---

- Pipelines/ tuberías:
    - Permiten redireccionar el canal de salida de un comando hacia el canal de entrada de otro.
    - El carácter que lo permite es : |  
\$ ls -l > resultado.txt  
\$ wc < resultdo.txt
- Aplicando una tubería:
- ```
$ ls -l | wc
```

## Comandos Básicos II

---

- Búsqueda de líneas:
  - Extracción de líneas de un fichero según varios criterios.
  - grep
  - egrep
  - fgrep

- Búsqueda de líneas:
  - **Grep**: Su nombre deriva de un comando en el editor de texto **ed** que tiene la siguiente forma: **g/re/p** y significa «hacer una búsqueda **g**lobal para las líneas que encajen con la **e**xpresión **r**egular (*regular expression* en inglés), e **im**primirlas **p**rint».
  - <http://www.esdebian.org/wiki/aprendiendo-expresiones-regularse-grep>

## Comandos Básicos II

---

- Búsqueda de líneas:
  - Fichero texto.txt
    - en un lugar de la mancha
    - # esto es un comentario
    - en un lugar del corazón
    - en un .....
    - en un lugar de la Mancha de cuyo nonmbre
    - última línea

## Comandos Básicos II

---

- Búsqueda de líneas:

- \$ egrep '^en un.\*mancha\$' texto.txt

- localiza líneas que comienzan por en un y termina por mancha.

- \$ egrep '^en un.\*[Mm]ancha\$' texto.txt

- \$ egrep '^[^#]\*' texto.txt

- Cualquier línea que no empiece por #

- \$ egrep '^#' texto

- Cualquier línea que empiece por #

- \b[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\b

- Búsqueda de IP's

## Comandos Básicos II

---

- Editor de flujo sed:
  - Permite filtrar y transformar texto.
  - Es como un editor de textos vía comandos, pero en un paso y sin edición interactiva.
  - Se utiliza muy a menudo para sustituir valores por otros o suprimir líneas particulares.
  - <http://www.gentoo.org/doc/es/articles/l-sed1.xml>



## Comandos Básicos II

---

- Editor de flujo sed:
  - Permite filtrar y transformar texto.
  - Es como un editor de textos vía comandos, pero en un paso y sin edición interactiva.
  - Se utiliza muy a menudo para sustituir valores por otros o suprimir líneas particulares.
  - Sintaxis de sustitución:
    - s/<antiguo>/nuevo/[g]
    - [g] opcional permite realizar la sustitución sobre toda la línea en caso de que haya varias coincidencias.

## Comandos Básicos II

---

- Editor de flujo sed:

```
$ echo 'Me llamo __NOMBRE__' | sed  
  's/__NOMBRE__/Juan/g'
```

```
$ route | sed "s/[ ][ ]*/;/g" > routetxt.txt
```

Sustituye los espacios en blanco por carácter ;

```
$ cat routetxt.txt | sed '1d' > fichero.txt
```

Elimina la primera línea de routetxt.txt

```
$ cat routetxt.txt | sed '1,10d' > fichero.txt
```

Elimina las diez primeras líneas de routetxt.txt

```
$ cat texto.txt | sed '/^$/d' > textosinblancos.txt
```

## Comandos Básicos II

---

- Columnas y Campos:
  - \$ cut -cColumnas [fichero..]
    - Una columnas es una posición de un carácter en una línea.
    - El primer carácter está en la columna 1, último en la columna 80.
    - \$ cut -c1 lista
    - \$ cut -c1-5,10-15 lista.txt

## Comandos Básicos II

---

- Columnas y Campos:
  - \$ cut -dCaracter Separador -fCampos [fichero..]
  - \$ cut -d\; -f1,3 fichero.txt

## Comandos Básicos II

---

- Recuento de líneas:
  - `$ wc [-l] [-c] [-w] [-m] fichero`
  - `-l` cuenta el número de líneas.
  - `-c` cuenta el número de bytes.
  - `-w` cuenta el número de palabras
  - `-m` cuenta el número de caracteres
  - `$ wc texto1.txt`
    - 2                      25                      125
    - 2 líneas            25 palabras    125 caracteres

## Comandos Básicos II

---

- Ordenación de líneas:
  - `$ sort [opciones] [-k pos1,[pos2]] [fichero]`
  - `sort` permite ordenar líneas.
  - Por defecto se realiza en toda la tabla en orden creciente.
  - Es posible especificar varios campos.
  - El separador por defecto es la tabulación o un espacio en blanco
  - Ejemplo: crea un fichero con 3 campos, nombre de producto, precio y cantidad.
  - `$ sort -n -r -k 3 productos.txt`

## Comandos Básicos II

---

- Ordenación de líneas:
  - Parámetros más usados:
    - -k    número de campo
      - -k1,3    -k1.3 ordenar por el tercer carácter del campo1
    - -d    Ordenación de diccionario, criterio de ordenación son las letras números y espacios.
    - -n    Ordenación numérica.
    - -b    Ignora espacios al principio del campo
    - -f    No diferencia entre mayúsculas y minúsculas.
    - -r    reverse, ordenación en orden decreciente
    - -tc    nuevo delimitador de campo c

## Comandos Básicos II

---

- Supresión de líneas repetidas:
  - `uniq` permite eliminar líneas repetidas en flujos de entrada o ficheros ordenados.
  - `$ cut -d: -f4 /etc/passwd | sort -n | uniq`



## Comandos Básicos II

---

- Sustitución de caracteres tr:
  - tr permite sustituir unos caracteres por otros
  - Únicamente acepta datos que provengan del canal estándar y no de ficheros.
  - \$ tr [opciones] original destino
  - \$ cat fichero.txt | tr “oi” “io”
  - \$ cat fichero.txt | tr “[a-z]” “[A-Z]”

## Comandos Básicos II

---

- Sustitución de caracteres tr:

### Eliminar repeticiones:

- `$ tr -s -d`
- `-s` sustituye conjunto de caracteres repetidos especificados por otro
- `$ tr ':' ' ' < /etc/passwd > ficheropasswd`
- `$ tr '[a-z]' '[A-Z]' < ficheropasswd`
- `$ tr ' ' '\n' < lineasusuarios`
- `$ tr -s ' ' < prueba2.txt`
  - En un lugar de la Mancha
  - En un lugar de la Mancha

## Comandos Básicos II

---

- Sustitución de caracteres tr:
  - -c : Hace que se traduzcan todos los caracteres que no se encuentren especificados en el primer parámetro. En el siguiente ejemplo se traduce por una ? todo lo que no sean letras o números.  
`$tr -c '[a-z][A-Z][0-9]' ? < fichero`
  - -d : Borra los caracteres que especifiquemos.
  - tr -d '[a-z][0-9]' < fichero
  - En un \$ lugar de la & Mancha de (

## Comandos Básicos II

---

- Sustitución de caracteres tr:

Trataremos de aislar la IP de

- `$ ifconfig eth0 | grep "inet "`
- `$ ifconfig eth0 | grep "inet " | tr -s " " ":"`
- `$ ifconfig eth0 | grep "inet " | tr -s " " ":" | cut -d: -f4`

## Comandos Básicos II

---

- Visualización de texto:
  - Página por página: pg, more, less
    - Opciones de less:
      - Barra espaciadora, avanza una página.
      - Intro, avanza una línea.
      - G, ir a final del texto.
      - G, ira al principio del texto
      - /palabra, busca la palabra avanzando en el texto
      - ?palabra, busca la palabra retrocediendo en el texto
      - N, busca la siguiente ocurrencia en el texto.
      - AvPág, RePág avanza o retrocede una página.
      - v, abre editor de texto
      - q, sale.

## Comandos Básicos II

---

- Visualización de texto:
  - `$ cat` visualización en bloque del fichero.
  - `$ cat -n` numeración de las líneas del fichero
  - `$ wc -l` cuenta las líneas del fichero.
  - `$ tac` visualización en bloque desde la última fila a la primera fila del fichero.

## Comandos Básicos II

---

- Principio de un fichero:
  - `$ head [-n] fichero`
  - Permite visualizar las n primeras líneas
  - Por defecto se visualizan las diez primeras
  - `$head -2 fichero`
- Final de un fichero:
  - `$ tail [-n] fichero`
  - Por defecto se visualizan las diez últimas

## Comandos Básicos II

---

- Comando tee:
  - Duplicamos la salida del canal estándar
  - `$ tee [-a] fichero`
  - `$ ls -l | tee midirectorio.txt`
    - Muestra por pantalla y crea el fichero con la misma información mostrada por pantalla.
    - Si el fichero no existe lo crea.
    - `[-a]`, permite sobrescribir.



## Comandos Básicos II

---

- División de ficheros en partes:
  - \$ split permite dividir un fichero en partes iguales de un tamaño determinado.
  - Podemos dividir el fichero por:
    - -l Recorte por líneas: los ficheros de salida tendrán todos el mismo número de líneas.
    - -b Recorte a tamaño fijo de n bytes.
      - nb: n bloques de 512 byte.
      - nk: n kilo bytes.
      - nm: n Mega bytes.
  - \$ split [-l n [-b n[bkm]] [fichero]
  - Tarea: divide un fichero mp3 en partes iguales.

## Comandos Básicos II

---

- Reconstruir ficheros divididos:
  - \$ cat fichero\* > nuevofichero