

# Computación en Física

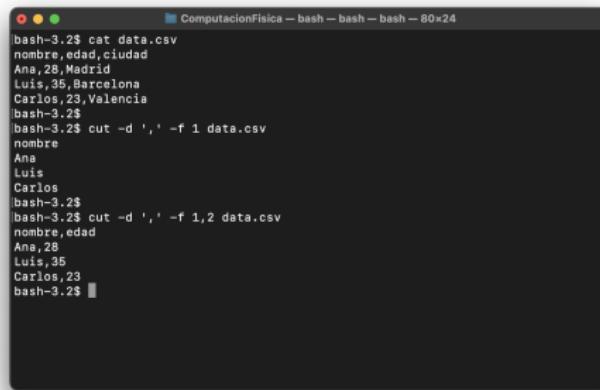
Harvey Rodriguez Gil

Universidad EIA

1 de Agosto de 2024

# Manipulación y Visualización de archivos - cut

cut es un comando en Linux que nos permite modificar la presentación de un archivo de datos tabulares o columnas específicas de un archivo separado por comas (csv).



The screenshot shows a terminal window titled "ComputacionFisica" with the command line "bash-3.2\$". It displays the contents of a CSV file named "data.csv" which contains three columns: nombre, edad, and ciudad. The first row shows "Ana,28,Madrid", the second "Luis,35,Barcelona", and the third "Carlos,23,Valencia". The user then runs the command "cut -d ',' -f 1 data.csv" to extract the first column, resulting in the output "nombre" followed by the names Ana, Luis, and Carlos. Finally, the command "cut -d ',' -f 1,2 data.csv" is run to extract the first two columns, resulting in the output "nombre,edad" followed by the pairs Ana,28, Luis,35, and Carlos,23.

```
bash-3.2$ cat data.csv
nombre,edad,ciudad
Ana,28,Madrid
Luis,35,Barcelona
Carlos,23,Valencia
bash-3.2$
bash-3.2$ cut -d ',' -f 1 data.csv
nombre
Ana
Luis
Carlos
bash-3.2$ cut -d ',' -f 1,2 data.csv
nombre,edad
Ana,28
Luis,35
Carlos,23
bash-3.2$
```

# Manipulación y Visualización de archivos - tr

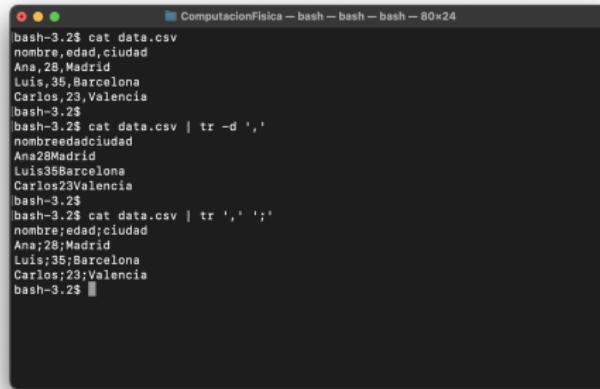
tr es un comando en Linux que nos permite remplazar o eliminar caracteres de la entrada estándar.

```
tr [opciones] SET1 [SET2]
```

Donde SET1 es el conjunto de caracteres a remplazar o eliminar y SET2 el conjunto de caracteres que remplazará a SET1

# Manipulación y Visualización de archivos - tr

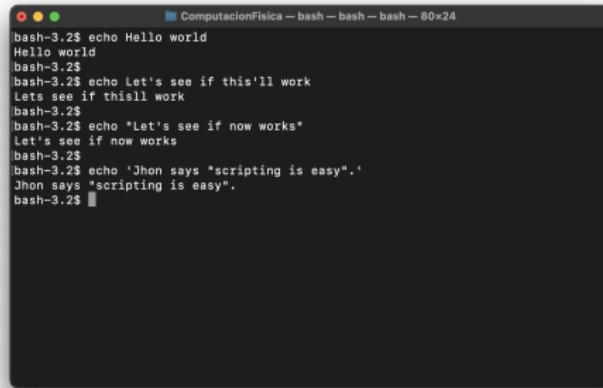
tr -d nos permite eliminar un carácter, si no es acompañado por este flag, se realizará un remplazo



```
ComputacionFisica — bash — bash — bash — 80x24
bash-3.2$ cat data.csv
nombre;edad;ciudad
Ana;28;Madrid
Luis;35;Barcelona
Carlos;23;Valencia
bash-3.2$
bash-3.2$ cat data.csv | tr -d ','
nombreedadciudad
Ana28Madrid
Luis35Barcelona
Carlos23Valencia
bash-3.2$
bash-3.2$ cat data.csv | tr ',' ';'
nombre;edad;ciudad
Ana;28;Madrid
Luis;35;Barcelona
Carlos;23;Valencia
bash-3.2$
```

# Presentando mensajes - echo

La mayoría de comandos producen sus propias salidas, como listar los objetos de un directorio con `ls`, o imprimir las primeras 10 líneas de un archivo con `head`. En el caso de requerir presentar un mensaje personalizado, podemos hacer uso del comando `echo`

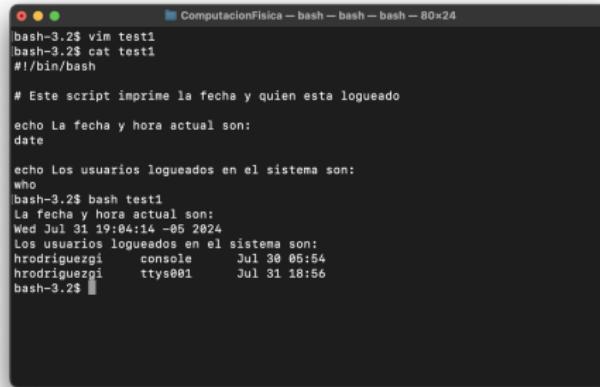


A screenshot of a macOS terminal window titled "ComputacionFisica – bash – bash – bash – 80x24". The window shows the following command history:

```
bash-3.2$ echo Hello world
Hello world
bash-3.2$ echo Let's see if this'll work
Let's see if this'll work
bash-3.2$ echo "Let's see if now works"
Let's see if now works
bash-3.2$ echo 'Jhon says "scripting is easy".'
Jhon says "scripting is easy".
bash-3.2$
```

# Presentando mensajes - echo

Este comando también puede ser utilizado en los scripts de bash:



```
ComputacionFisica -- bash -- bash -- bash -- 80x24
bash-3.2$ vim test1
bash-3.2$ cat test1
#!/bin/bash

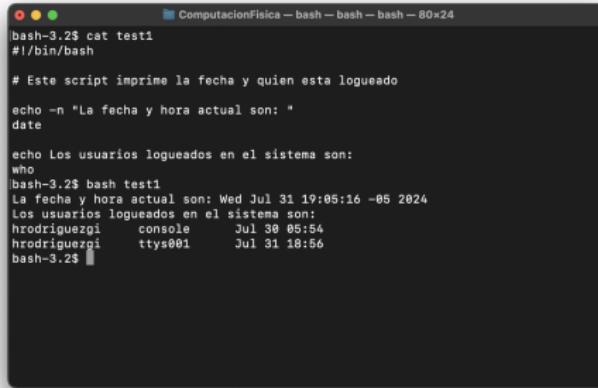
# Este script imprime la fecha y quien esta logueado

echo La fecha y hora actual son:
date

echo Los usuarios logueados en el sistema son:
who
bash-3.2$ bash test1
La fecha y hora actual son:
Wed Jul 31 19:04:14 -05 2024
Los usuarios logueados en el sistema son:
hrodriguezgi    console      Jul 30 06:54
hrodriguezgi    ttys001     Jul 31 18:56
bash-3.2$
```

# Presentando mensajes - echo

Y con el flag `-n` podemos modificar el comportamiento de la salida, eliminando el carácter `\n` que significa salto de linea:



```
ComputacionFisica — bash — bash — bash — 80x24
bash-3.2$ cat test1
#!/bin/bash

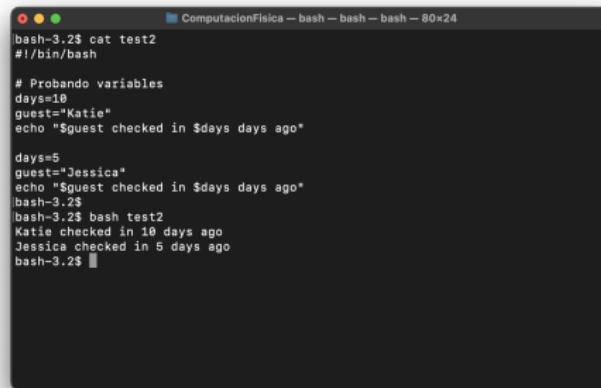
# Este script imprime la fecha y quien esta logueado

echo -n "La fecha y hora actual son: "
date

echo Los usuarios logueados en el sistema son:
who
bash-3.2$ bash test1
La fecha y hora actual son: Wed Jul 31 19:05:16 -05 2024
Los usuarios logueados en el sistema son:
hrodriguezpi    console      Jul 30 06:54
hrodriguezpi    ttys001     Jul 31 18:56
bash-3.2$
```

# Variabes de usuario

Adicional a las variables Globales y Locales, también tenemos las de usuarios que son utilizadas en los scripts. Esto nos permite personalizar aún más los scripts de bash para desarrollar tareas cada vez más complejas



The screenshot shows a terminal window titled "ComputacionFisica" with the command "bash - 80x24". The script content is as follows:

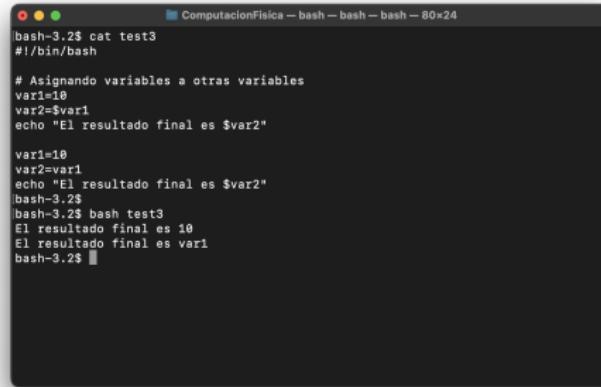
```
bash-3.2$ cat test2
#!/bin/bash

# Probando variables
days=10
guest="Katie"
echo "$guest checked in $days days ago"

days=5
guest="Jessica"
echo "$guest checked in $days days ago"
bash-3.2$
bash-3.2$ bash test2
Katie checked in 10 days ago
Jessica checked in 5 days ago
bash-3.2$
```

# Variabes de usuario

Las variables pueden ser asignadas a otras variables, lo que se debe tener en cuenta es el símbolo de \$, ya que de no ponerlo, podría generar salidas no deseadas



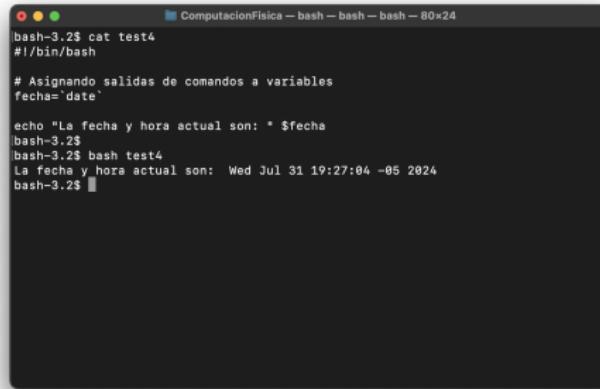
```
ComputacionFisica -- bash -- bash -- bash - 80x24
bash-3.2$ cat test3
#!/bin/bash

# Asignando variables a otras variables
var1=10
var2=$var1
echo "El resultado final es $var2"

var1=10
var2=var1
echo "El resultado final es $var2"
bash-3.2$ bash test3
El resultado final es 10
El resultado final es var1
bash-3.2$
```

# Variabes de usuario - backtick

El backtick ‘ es comunmente utilizado en los scripts de bash para asignar a variables salidas de los comandos propios de linux. Este caracter deberá estar al inicio y al final del comando que queremos almacenar en una variable



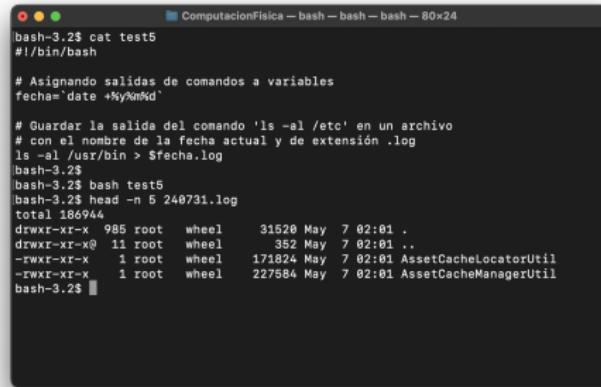
```
ComputacionFisica -- bash -- bash -- bash -- 80x24
bash-3.2$ cat test4
#!/bin/bash

# Asignando salidas de comandos a variables
fecha=`date`

echo "La fecha y hora actual son: " $fecha
bash-3.2$ bash test4
La fecha y hora actual son: Wed Jul 31 19:27:04 -05 2024
bash-3.2$
```

# Variabes de usuario - backtick

El comando que quiero ejecutar y que cuenta con opciones o flags debe estar también encerrado entre los backticks



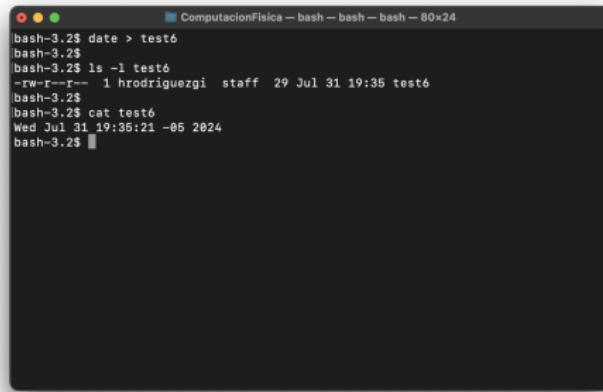
```
ComputacionFisica -- bash -- bash -- bash - 80x24
bash-3.2$ cat test5
#!/bin/bash

# Asignando salidas de comandos a variables
fecha=`date +%y%m%d`

# Guardar la salida del comando 'ls -al /etc' en un archivo
# con el nombre de la fecha actual y de extensión .log
ls -al /usr/bin > $fecha.log
bash-3.2$
bash-3.2$ bash test5
bash-3.2$ head -n 5 240731.log
total 186944
drwxr-xr-x  985 root    wheel      31520 May  7 02:01 .
drwxr-xr-x@ 11 root    wheel       362 May  7 02:01 ..
-rw-r--r--@  1 root    wheel     171824 May  7 02:01 AssetCacheLocatorUtil
-rw-r--r--@  1 root    wheel     227584 May  7 02:01 AssetCacheManagerUtil
bash-3.2$
```

# Redirección de salida

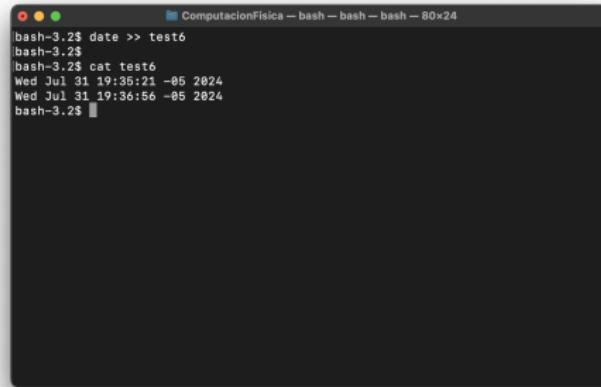
La forma más basica de redireccionar la salida de un comando a un archivo es por medio del caracter >



```
ComputacionFisica - bash - bash - bash - 80x24
bash-3.2$ date > test6
bash-3.2$
bash-3.2$ ls -l test6
-rw-r--r-- 1 hrodriguezgi staff 29 Jul 31 19:35 test6
bash-3.2$
bash-3.2$ cat test6
Wed Jul 31 19:35:21 -05 2024
bash-3.2$
```

# Redirección de salida

Sin embargo se debe tener en cuenta que esto hace que el archivo se sobre escriba cada vez. Si en lugar de esto, queremos ir almacenando la salida de diferentes comandos, se deberá utilizar >>

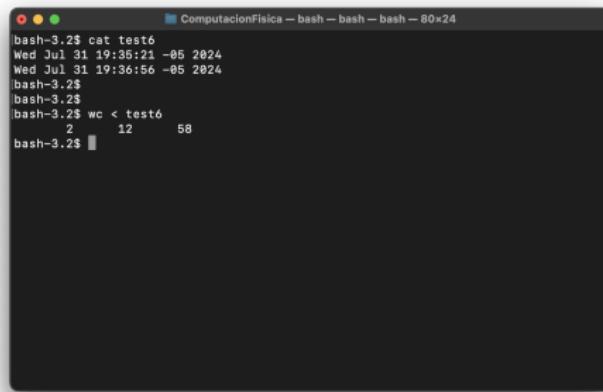


A screenshot of a Linux terminal window titled "ComputacionFisica — bash — bash — bash — 80x24". The terminal shows the following session:

```
bash-3.2$ date >> test6
bash-3.2$ 
bash-3.2$ cat test6
Wed Jul 31 19:35:21 -05 2024
Wed Jul 31 19:36:56 -05 2024
bash-3.2$ 
```

# Redirección de entrada

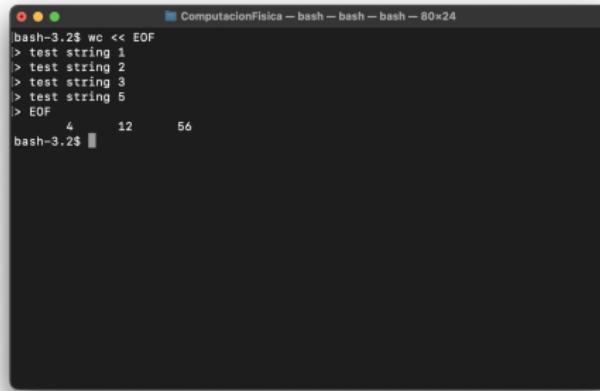
La redirección de entrada toma el contenido de un archivo y lo redirecciona a un comando (lo opuesto a la redirección de salida). La redirección de entrada se realizará con el carácter <



```
ComputacionFisica — bash — bash — bash — 80x24
bash-3.2$ cat test6
Wed Jul 31 19:35:21 -05 2024
Wed Jul 31 19:36:56 -05 2024
bash-3.2$ 
bash-3.2$ 
bash-3.2$ wc < test6
      2      12      58
bash-3.2$
```

# Redirección de entrada

Esto también puede ser utilizado para enviar múltiples líneas desde la consola a un comando, solo que esta vez se utilizará <<



A screenshot of a terminal window titled "ComputacionFisica - bash - bash - bash - 80x24". The window contains the following text:

```
bash-3.2$ wc << EOF
> test string 1
> test string 2
> test string 3
> test string 5
> EOF
      4      12      56
bash-3.2$
```

# Pipes

Hay ocasiones en los que no queremos enviar la salida de un comando a un archivo (>), si no que queremos pasarlo a otro comando para hacer algo más complejo.

```
hrodriguezgi - root@cb03401b9885: / - docker run -it ubuntu /bin/bash - 80x24
root@cb03401b9885: # dpkg -l > dpkg.list
root@cb03401b9885: # sort < dpkg.list
=====
=====
Desired=Unknown/Install/Remove/Purge/Hold
ii apt 2.7.14build2 arm64 commandline pack
age manager
ii base-files 13ubuntu10 arm64 Debian base syst
em miscellaneous files
ii base-passwd 3.6.3build1 arm64 Debian base syst
em master password and group files
ii bash 5.2.21-2ubuntu4 arm64 GNU Bourne Again
Shell
ii bsdtutils 1:2.39.3-9ubuntu6 arm64 basic utilities
from 4.4BSD-Lite
ii coreutils 9.4-3ubuntu6 arm64 GNU core utiliti
es
ii dash 0.5.12-6ubuntu5 arm64 POSIX-compliant
shell
ii debconf 1.5.8ubuntu1 all Debian configura
tion management system
ii debianutils 5.17build1 arm64 Miscellaneous ut
ilities specific to Debian
ii diffutils 1:3.10-1build1 arm64 File comparison
```

# Pipes

Linux cuenta con pipe |, el cual nos permite redireccionar la salida de un comando como argumento para el siguiente comando, en otras palabras podemos unir dos o más comandos.

```
root@cb03401b9885:/# dpkg -l | sort
=====
Desired=Unknown/Install/Remove/Purge/Hold
ii apt                    2.7.14build2      arm64      commandline pack
age manager
ii base-files              13ubuntu10      arm64      Debian base syst
em miscellaneous files
ii base-passwd              3.6.3build1      arm64      Debian base syst
em master password and group files
ii bash                   5.2.21-2ubuntu4    arm64      GNU Bourne Again
Shell
ii bsdutils                1:2.39.3-9ubuntu6    arm64      basic utilities
from 4.4BSD-Lite
ii coreutils                9.4-ubuntu6       arm64      GNU core utiliti
es
ii dash                   0.5.12-6ubuntu5     arm64      POSIX-compliant
shell
ii debconf                 1.5.8ubuntu1      all       Debian configura
tion management system
ii debianutils              5.17build1      arm64      Miscellaneous ut
ilities specific to Debian
ii diffutils                1:3.10-1build1    arm64      File comparison
utilities
```

# Pipes

En el siguiente ejemplo veremos como se unen tres comandos: `dpkg -l` que lista los programas instalados, `sort` que ordena alfabeticamente la salida y por ultimo `more` para imprimir por páginas el resultado

```
root@cb03401b9885:/# dpkg -l | sort | more
=====
Desired=Unknown/Install/Remove/Purge/Hold
ii apt                    2.7.1ubuntu2      arm64      commandline pack
age manager
ii base-files              13ubuntu10     arm64      Debian base syst
em miscellaneous files
ii base-passwd              3.6.3build1    arm64      Debian base syst
em master password and group files
ii bash                   5.2.21-2ubuntu4   arm64      GNU Bourne Again
SHell
ii bsutils                 1:2.39.3-9ubuntu6   arm64      basic utilities
from 4.4BSD-Lite
ii coreutils                9.4-3ubuntu6     arm64      GNU core utiliti
es
ii dash                   0.5.12-6ubuntu5   arm64      POSIX-compliant
shell
ii debconf                  1.5.86ubuntu1    all        Debian configura
tion management system
ii debianutils               5.17build1     arm64      Miscellaneous ut
ilities specific to Debian
ii diffutils                1:3.10-1build1   arm64      File comparison
utilities
```

# Vamos a practicar

Descargar el archivo `clase_8.tar.gz` del repositorio de GitHub y seguir las instrucciones

