Programación con Shell Scripting: Sesión 8

Máster Universitario en Bioinformática

Universidad Internacional de Valencia

Dra. Paula Soler Vila
paula.solerv@professor.universidadviu.com

De:
Planeta Formación y Universidades

Universidad Internacional de Valencia

Aspectos a tratar

1. Ejecución de un script
2. Variables de entorno global: **PATH**
3. Variables especiales
4. Ejercicios varios: **REPASO**
5. Trabajando con variables

Cómo ejecutar un script

```
#!/usr/bin/bash
#
##Fecha de escritura: 13/10/2022
##Este programa te informa en que directorio se encuentra trabajando.
#
lugar=$(pwd)
echo "en este momento te encuentras trabajando en este $lugar, no vayas a perderte"
~
```

location.sh

```
$ ls
Asignaturas Documents gencoce_annotation_v44 location.sh MultiQC Pictures R Videos Desktop Downloads
gencode_annotation_v45 miniconda3 Music Public Templates


$ location.sh
bash: location.sh: command not found
```

Comandos y PATH

```
$ which pwd
 /usr/bin/pwd


$ echo $PATH
/home/paula.soler/miniconda3/bin:/home/paula.soler/miniconda3/condabin:/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local
/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin
```

- Ubicaciones donde la Shell buscará un comando (programa ejecutable).
- Los nombres de los directorios se escriben usando rutas absolutas.
- Los dos puntos se emplean como delimitador.

- Búsqueda **secuencial**
- De no estar en ninguno de los directorios listados entonces se producirá un mensaje de error

Comandos y PATH

```
$ location.sh
bash: location.sh: command not found
```

Mover nuestro script a uno de los directorios del **PATH**

Añadir nuestro directorio actual a **PATH**

Cambiar la forma en la que intentamos llamar a nuestro script

Ejecución **explícita**

En estas formas, tanto en bash location como en . location.sh hay que dejar un espacio en blanco entre ellas

```
$ cat location.sh
#!/usr/bin/bash #
##Fecha de escritura: 13/10/2022
##Este programa te informa en que directorio se encuentra trabajando la terminal.
lugar=$(pwd)
echo "En este momento te encuentras trabajando en este $lugar, no vayas a perderte"

$ bash location.sh
En este momento te encuentras trabajando en este /home/paula.soler, no vayas a perderte
```

Ejecución **implícita**

```
$ . location.sh
En este momento te encuentras trabajando en este /home/paula.soler, no vayas a perderte

$ ./location.sh
bash: ./location.sh: Permission denied
$ chmod u+x location.sh
$ ./location.sh
En este momento te encuentras trabajando en este /home/paula.soler, no vayas a perderte
```

Ejecución **explícita**

```
$ cat location.sh
#!/usr/bin/bash #
##Fecha de escritura: 13/10/2022
##Este programa te informa en que directorio se encuentra trabajando la terminal.
lugar=$(pwd)
echo "En este momento te encuentras trabajando en este $lugar, no vayas a perderte"

$ bash location.sh
En este momento te encuentras trabajando en este /home/paula.soler, no vayas a perderte
```

Ejecución **implícita**

```
$ . location.sh
En este momento te encuentras trabajando en este /home/paula.soler, no vayas a perderte

$ ./location.sh
bash: ./location.sh: Permission denied
$ chmod u+x location.sh
$ ./location.sh
En este momento te encuentras trabajando en este /home/paula.soler, no vayas a perderte
```

Scripts en la **subshell**

```
$ cat script.sh
#!/usr/bin/bash
var=LHB
echo $var
```

=

```
$ cat script.sh
#!/usr/bin/bash
var=LHB ; echo $var
```

```
# Eliminar la variable definida del interior del script
$ cat script.sh
#!/bin/bash
echo $var


# Determinar la variable local en nuestra Shell principal
$ var=LHB
```

Scripts en la **subshell**

```
$ cat script.sh
#!/bin/bash
echo $var
```

```
$ var=LHB
$ bash script.sh
```

Explícita

```
$ . script.sh
LHB
$ ./script.sh
```

Implícita

Los scripts se ejecutan por defecto en una sub-shell a excepción de si utilizamos el carácter << . >>

Pero con ./ va a pasar igual que de la forma explícita con bash, que va a ocurrir en la subshell y la variable local está en la shell principal, por tanto no va a salir

Comandos y PATH

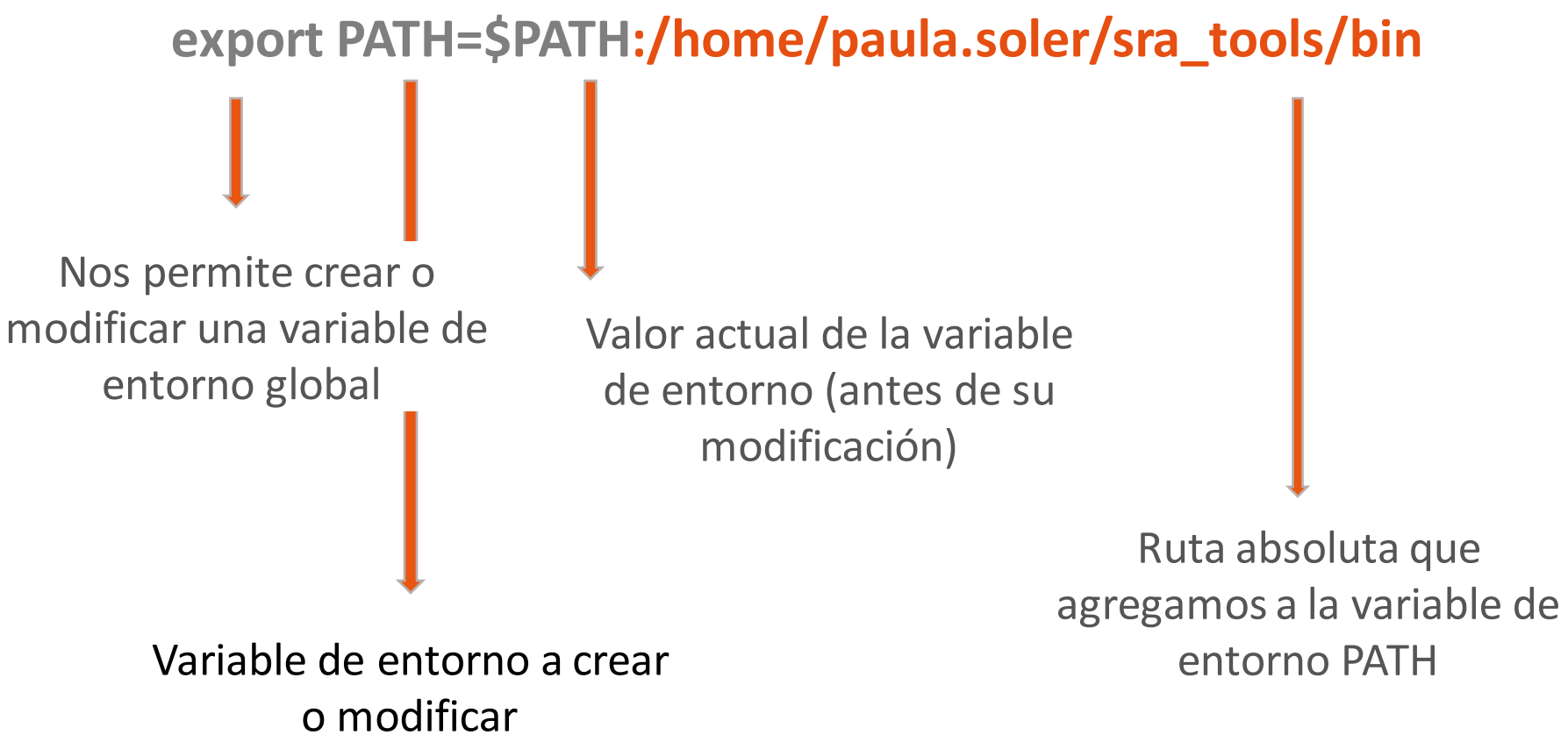
```
$ location.sh
bash: location.sh: command not found
```

Mover nuestro script a uno de los directorios del **PATH**

Añadir nuestro directorio actual a **PATH**

Cambiar la forma en la que intentamos llamar a nuestro script

¿Cómo agregamos un directorio al PATH?



¿Cómo agregamos un directorio al PATH?

```
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin

$ export PATH=$PATH:/home/paula.soler/sra_tools/bin

$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin:/home/paula.soler/sra_tools/bin
```

```
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin

$ export PATH=/home/paula.soler/sra_tools/bin:$PATH

$ echo $PATH
home/paula.soler/sra_tools/bin:/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin
```

¿Qué sucede si reiniciamos nuestra sesión o abrimos otra terminal?

- La **modificación** realizada en la variable PATH se **pierde.**

- Cada vez que el **Shell se inicia**, se establecen las variables de entorno global, incluyendo PATH, pero se puede configurar para que siempre incluya su nueva ruta con cada nuevo Shell que abra.

- La forma exacta de hacerlo depende del Shell que esté ejecutando.

¿Qué sucede si reiniciamos nuestra sesión o abrimos otra terminal?

Para Bash, simplemente agregamos

export PATH=$PATH:/home/paula.soler/sra_tools/bin

en el archivo apropiado que es leído cuando se inicia el Shell

~/.bash_profile

~/.bashrc

~/.profile

cuando se trabaja en remoto se suele recomendar más usar la de .bashrc

El punto está diciendo que son archivos ocultos

¿Dónde se encuentra el archivo .bashrc?

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ pwd
/home/paula.soler
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ ls -a

.           .bash_logout   .cache   Desktop   especiales.sh   .ICEauthority   .lesshst   MultiQC   Public   .Rhistory   Videos          .zshrc
..          .bash_profile  .conda   Documents .face           .icons          .local     Music     .purple  .swp        .viminfo
Asignaturas .bashrc        .config  Downloads .gitconfig      impresion.sh    miniconda3 .ncbi     .r       Templates   .Xauthority
.bash_history .bashrc.back .dbus    .esd_auth .gstreamer-0.10 .java           .mozilla   Pictures  R        .themes     .xsession-errors

(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
```

¿Cómo agregamos un directorio de forma PERMANENTE al PATH?

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo "export PATH=$PATH:/RUTA_NUEVA" >> /home/paula.soler/.bashrc

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat /home/paula.soler/.bashrc

# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export
PATH=/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin:/RUTA_NUEVA
```

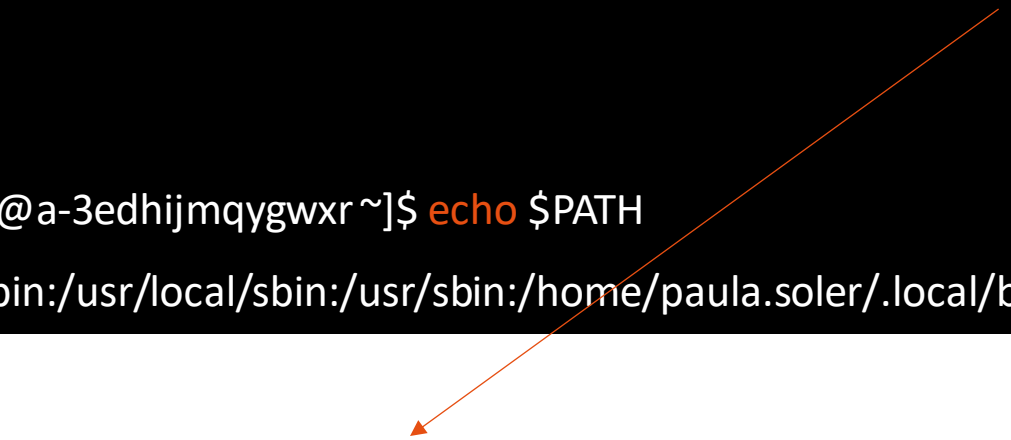
¿Cómo agregamos un directorio de forma PERMANENTE al PATH?

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin
```

Con el comando source se actualiza y se ejecuta, no haría falta reiniciar

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ source /home/paula.soler/.bashrc
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/sbin:/usr/local/sbin:/usr/sbin:/home/paula.soler/.local/bin:/home/paula.soler/bin:/RUTA_NUEVA
```



```
$ cp ~/.bashrc ~/.bashrc.back
```

es recomendable hacer una copia con el comando cp, porque son cambios permanentes y si pasa algo asi podemos ir al archivo original.

Universidad Internacional de Valencia

Aspectos a tratar

1. Ejecución de un script
2. Variables de entorno global: **PATH**
3. Variables especiales
4. Ejercicios varios: **REPASO**
5. Trabajando con variables

Variable Especiales

```
#!/usr/bin/bash

echo "Filename : $0"
echo "First Argument : $1"
echo "Second Argument : $2"
echo "Total no. of args: $#"


echo $SECONDS
echo $RANDOM
echo $LINENO
```

Nos permite contabilizar el tiempo que tarda en escribir

nos permite generar un número aleatorio cada vez que lancemos ese script

Indica el número de lineas que tiene el script

special_variables.sh

Vim es un editor de texto

Para empezar a escribir hay que darle a I

Para salir hay que darle a la tecla esc y después :x que permite guardar y salir o :wq

PRACTIQUEMOS

Variable Especiales

```
#!/usr/bin/bash

echo "Filename : $0"
echo "First Argument : $1"
echo "Second Argument : $2"
echo "Total no. of args: $#"


echo $SECONDS
echo $RANDOM
echo $LINENO
```

special_variables.sh

En este caso al bash no le hacen falta permisos de ejecución

echo "First Argument : $1“ (del 1 al 9) -> ${10} ${11}

Cuando se leen los scripts se leen en orden secuencialmente (contando espacios) y leerá hasta donde pone LINENO

Universidad Internacional de Valencia

Aspectos a tratar

1. Ejecución de un script
2. Variables de entorno global: **PATH**
3. Variables especiales
4. Ejercicios varios: **REPASO**
5. Trabajando con variables

Repaso de las variables: Enlaza las instrucciones con su resultado

Intrucciones **PREGUNTA DE EXAMEN** **Resultados**

> variable=Hola

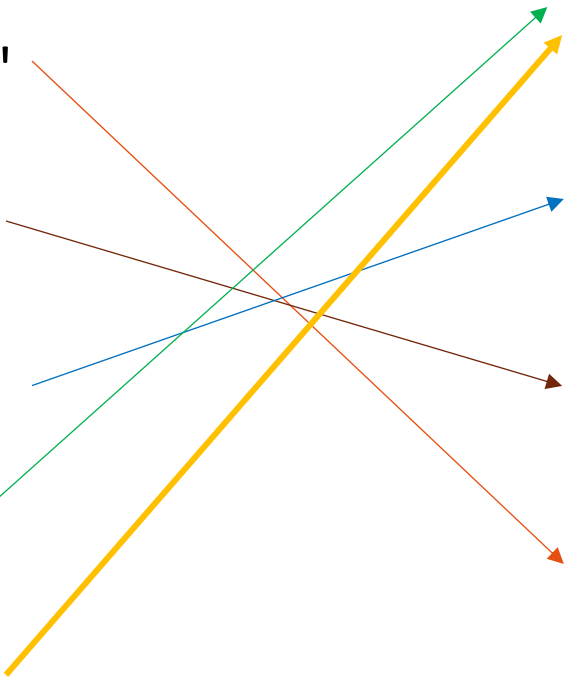
> echo "$variable mundo"

> echo '$variable mundo'

> echo `$variable mundo`

> echo `echo $variable`

> echo $(echo $variable)



1. Hola
2. bash: Hola: command not found

el acento grave era para expandir comandos, pero hola no es ningún comando

3. $variable mundo
4. Hola mundo

Repaso de las variables: Enlaza las instrucciones con su resultado

Instrucciones

>A=100
>B=200

>echo "A:$A | B:$B | C:$[A + B]"
>echo "A:$A | B:$B | C:$((A + B))"

Resultados

A: 100 | B: 200 | C:300

> echo "teclea una palabra"
> **read** palabra
> echo "la palabra tecleada es $palabra"

Teclee una palabra

(usuario teclea una palabra)

La palabra tecleada es **dinámica**

Comando read

El comando **read** lee la entrada estándar y asigna las palabras leídas en la(s) variable(s) cuyo nombre se pasa como argumento.

```
read var1 var2 …
```

Ejemplo

```
$ read var1
 hola
$ echo $var1
hola
```

La palabra introducida se almacena en la variable **var1**

Comando read (*History*)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ read var1
hola
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $var1
hola
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ read var1
Estoy testeando el comando read
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $var1
Estoy testeando el comando read
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ read var1 var2
genoma humano
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $var1
genoma
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $var2
humano
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ read var1 var2
Estoy testeando el comando read
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $var1
Estoy
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $var2
testeando el comando read
```

palabra.sh

```
#!/bin/bash
echo "teclea una palabra"
read palabra
echo "la palabra tecleada es $palabra"
```

Universidad Internacional de Valencia

Aspectos a tratar

1. Ejecución de un script
2. Variables de entorno global: **PATH**
3. Variables especiales
4. Ejercicios varios: **REPASO**
5. Trabajando con variables

Trabajar con variables en Bash

1. Encontrar la longitud de la cadena

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ variable1=paula
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $variable1
paula

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo $variable1 | wc -m
6

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo -n $variable1 | wc -m
5

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${#variable1}
5
```

la opción echo -n elimina el salto de línea, por lo que ya se podría usar wc -m que cuenta caracteres pero con saltos de lína incluído

OJO, IMP!! simplemente añadiendo # puedes acceder al número de caracteres

2. Extracción de subcadenas

${cadena:posición_caracter:longitud}

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ var="uno;dos;tres“
```

01234

123

¿Cómo extraemos la subcadena “dos”?

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var:5:2}
os

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var:4:3}
dos
```

Empezamos a contar desde 0 (índice 0)

2. Extracción de subcadenas

${cadena:posición_caracter:longitud}

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ var="uno;dos;tres“
```

01234 → 12345678

¿Cómo extraemos la subcadena dos;tres?

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var:4:8}
dos;tres

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var:4}
dos;tres
```

2. Extracción de subcadenas

```
${cadena:posición_caracter:longitud}
```

string=ABCDEFGHIJKL

```
echo ${string:0} : ABCDEFGHIJKL
echo ${string:0:1} : A
echo ${string:3} : DEFGHIJKL
echo ${string:3:4} : DEFG
```

Si solo ponemos el 0 y no indicamos nada más, extrae la cadena entera

Porque cuenta desde el 0 y a partir de ahí solo un carácter, por tanto solo la A

Reemplazar subcadenas

echo ${variable/subcadena/reemplazo}

${cadena/buscar/reemplazar}
Sustituye la primera coincidencia de buscar con reemplazar

${cadena//buscar/reemplazar}
Sustituye todas las coincidencias de buscar con reemplazar

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ var="uno;dos;tres;uno"

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var/uno/cuatro}
cuatro;dos;tres;uno

[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var//uno/cuatro}
cuatro;dos;tres;cuatro
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ var="image.png"
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo ${var/png/jpg}
image.jpg
```

Universidad
Internacional
de Valencia

universidadviu.com

De:
Planeta Formación y Universidades