

Informática II

La shell de Linux (3/4)

Gonzalo F. Perez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2024 –

Redirección de entrada, salida y error

Redirección de salida (`stdout`)

► `ls -l > lsoutput.txt`

Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`

Redirección de entrada, salida y error

Redirección de salida (`stdout`)

- ▶ `ls -l > lsoutput.txt`

Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`

- ▶ `ps >> lsoutput.txt`

Agrega la salida del comando `ps` al archivo `lsoutput.txt`

Redirección de entrada, salida y error

Redirección de salida (`stdout`)

- ▶ `ls -l > lsoutput.txt`

Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`

- ▶ `ps >> lsoutput.txt`

Agrega la salida del comando `ps` al archivo `lsoutput.txt`

- ▶ Otro ejemplo:

`cat lsoutput.txt > /dev/null`

Redirección de entrada, salida y error

¿Dónde muestra la información los comandos de la Shell?

Redirección de entrada, salida y error

¿Dónde muestra la información los comandos de la Shell? → en la salida estándar.

Redirección de entrada, salida y error

¿Dónde muestra la información los comandos de la Shell? → en la salida estándar.

Todo proceso tiene abierto tres descriptores de archivos:

- ▶ 0: entrada estándar (`stdin`)
- ▶ 1: salida estándar (`stdout`)
- ▶ 2: error estándar (`stderr`)

Redirección de entrada, salida y error

Redirección de salida y error (`stdout` y `stderr`)

► `ls -ld /tmp /tnt`

Redirección de entrada, salida y error

Redirección de salida y error (`stdout` y `stderr`)

► `ls -ld /tmp /tnt`

► `ls -ld /tmp /tnt >/dev/null`

Redirección de entrada, salida y error

Redirección de salida y error (`stdout` y `stderr`)

- ▶ `ls -ld /tmp /tnt`
- ▶ `ls -ld /tmp /tnt >/dev/null`
- ▶ `ls -ld /tmp /tnt 2>/dev/null`

Redirección de entrada, salida y error

Ejemplo de uso de stdout y stderr

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void) {
5     int num;
6
7     do {
8         printf("Ingrese un nro entero (0 para salir): ");
9         scanf("%d", &num);
10        if(num == 0)
11            break; /* Ejemplo del uso de break */
12
13        if(num < 0)
14            fprintf(stderr, "ERROR: %d es un número negativo!\n", num);
15        else
16            fprintf(stdout, "La raíz cuadrada de %d es %g\n", num, sqrt(num));
17    } while(num != 0);
18
19    return 0;
20 }
```

Redirección de entrada, salida y error

Redirección de entrada (`stdin`)

► `ls -lR /usr/include > lsoutput.txt`

Redirección de entrada, salida y error

Redirección de entrada (`stdin`)

► `ls -lR /usr/include > lsoutput.txt`

► `more < lsoutput.txt`

Pipes

Se pueden conectar procesos utilizando el operador pipe (tuberías), |.

Pipes

Se pueden conectar procesos utilizando el operador pipe (tuberías), |.

Ejemplo: utilizar `sort` para ordenar de forma inversa la salida de `ls`

```
ls -l | sort -r
```

(Ver el manual de `ls` y `sort`)

Pipes

Se pueden conectar procesos utilizando el operador pipe (tuberías), `|`.

Ejemplo: utilizar `sort` para ordenar de forma inversa la salida de `ls`

```
ls -l | sort -r
```

(Ver el manual de `ls` y `sort`)

Si no se utiliza pipes, se necesitan varios pasos:

1. `ls -l > ls.txt`
2. `sort -r ls.txt > lsrev.txt`

Pipes

Se pueden conectar procesos utilizando el operador pipe (tuberías), `|`.

Ejemplo: utilizar `sort` para ordenar de forma inversa la salida de `ls`

```
ls -l | sort -r
```

(Ver el manual de `ls` y `sort`)

Si no se utiliza pipes, se necesitan varios pasos:

1. `ls -l > ls.txt`
2. `sort -r ls.txt > lsrev.txt`

Conectando los procesos mediante pipes:

```
► ls -l | sort -r > lsrev.txt
```

