



# Sistemas Operativos (75.08)

Segundo cuatrimestre 2022 - FIUBA

---

# Repositorio

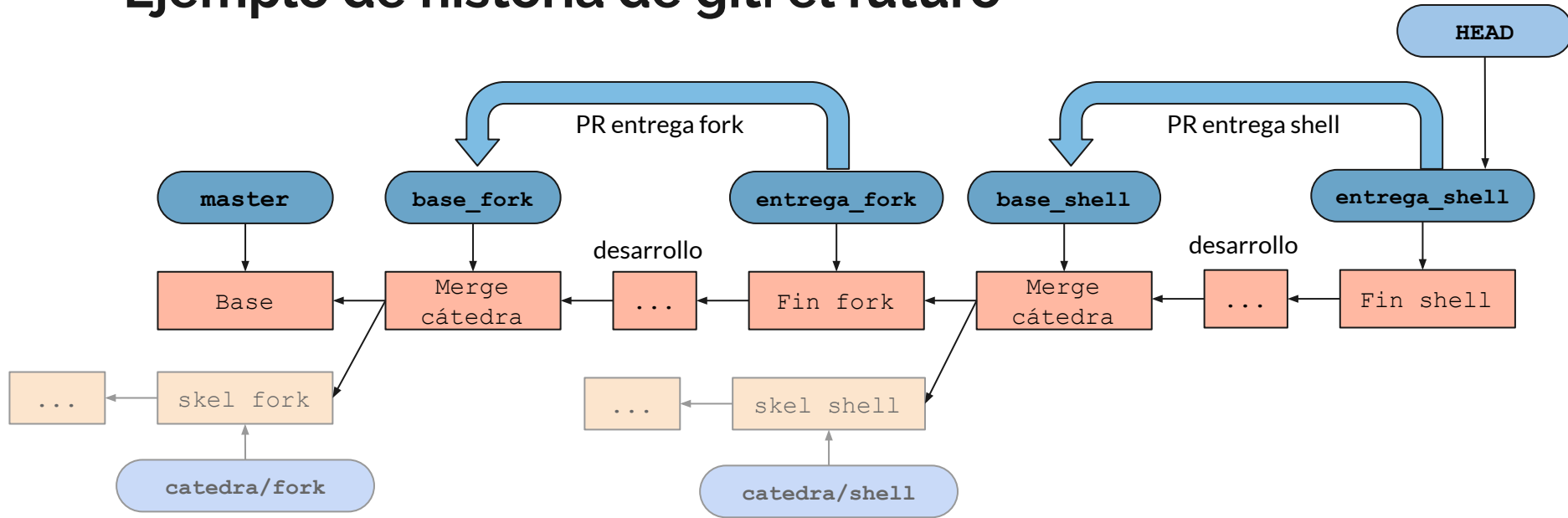


## Preparación repositorio

- `git checkout entrega_fork //` o también `git checkout main`
- `git checkout -b base_shell` (crea la rama “base\_shell”)
- `git fetch catedra`
- `git merge catedra/shell` (aparece el *merge commit*)
- `git push -u origin base_shell`
- `git checkout -b entrega_shell` (crea la rama “entrega\_shell”)
- `git push -u origin entrega_shell`

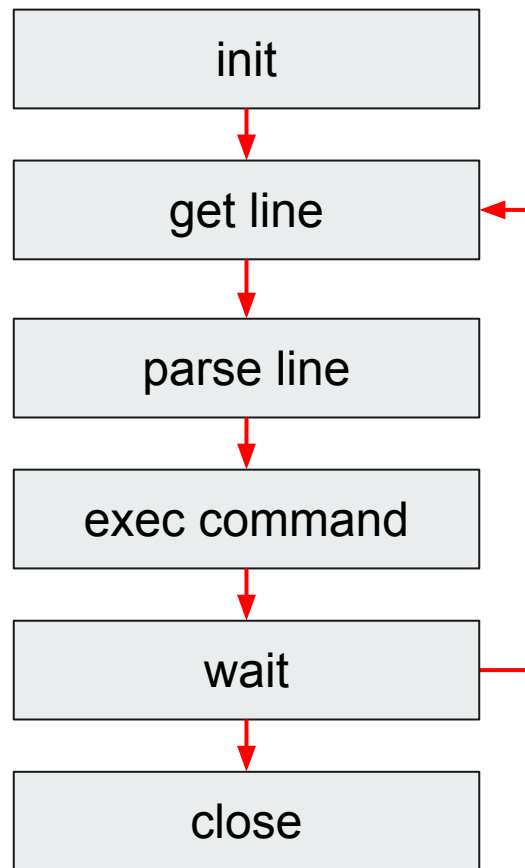
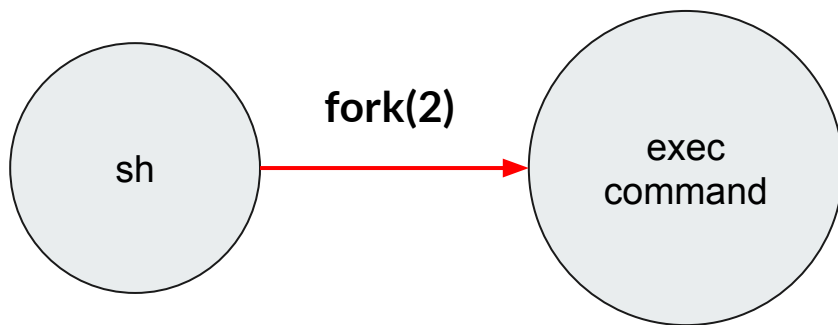
**IMPORTANTE:** todos los *commits* van en la rama *entrega\_shell*

# Ejemplo de historia de git: el futuro



---

# Lab shell

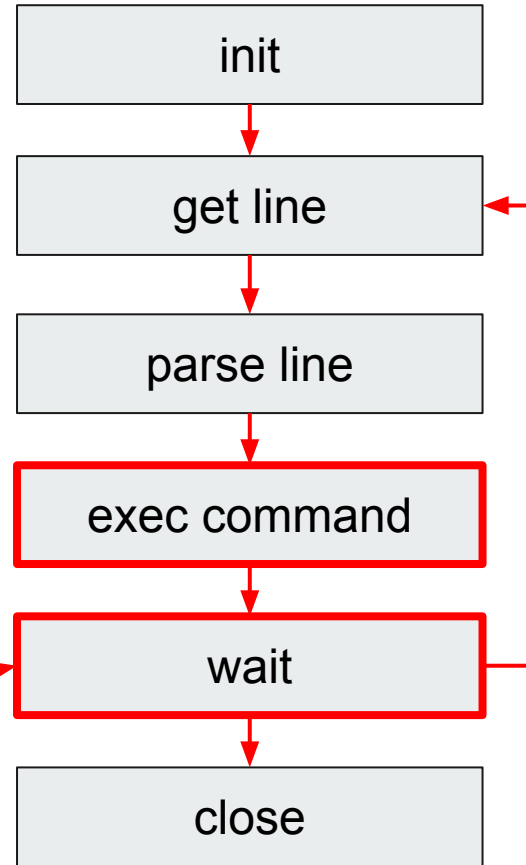


## Parte 1: Invocación de comandos

Syscalls:

- `fork(2)`
- `exec(2)`
- `wait(2)`

Cómo se puede hacer  
para tener un proceso en  
segundo plano?

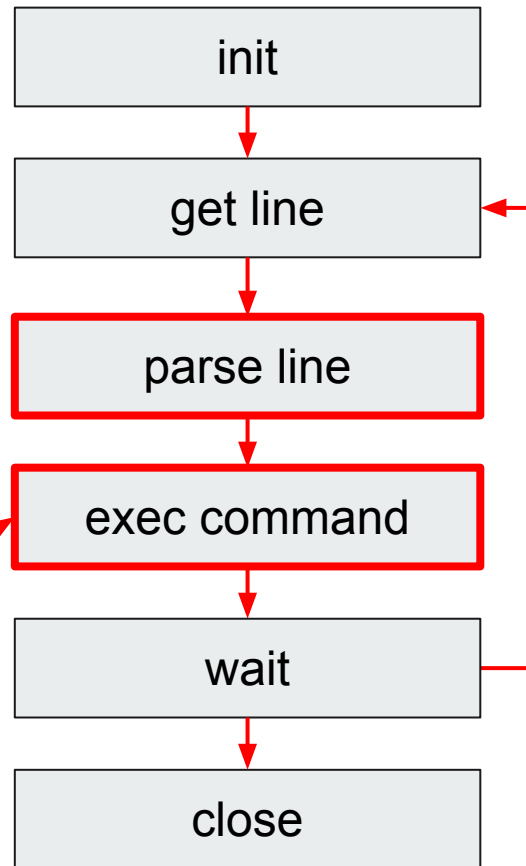


## Parte 2: Redirecciones (flujo estándar)

Syscalls:

- `dup(2)`
- `open(2)`
- `close(2)`

Cómo se puede hacer  
para redireccionar  
*stdin/stdout* hacia un  
archivo específico?





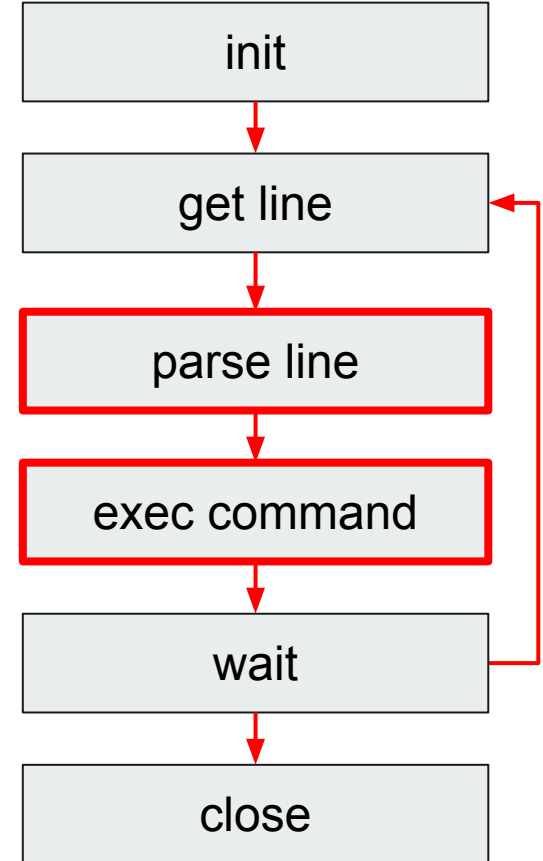
## Parte 2: Redirecciones (flujo estándar)

### Prueba 0:

1. `close(1)`
2. `open("file-out.txt")`

O también:

1. `close(0)`
2. `open("file-in.txt")`



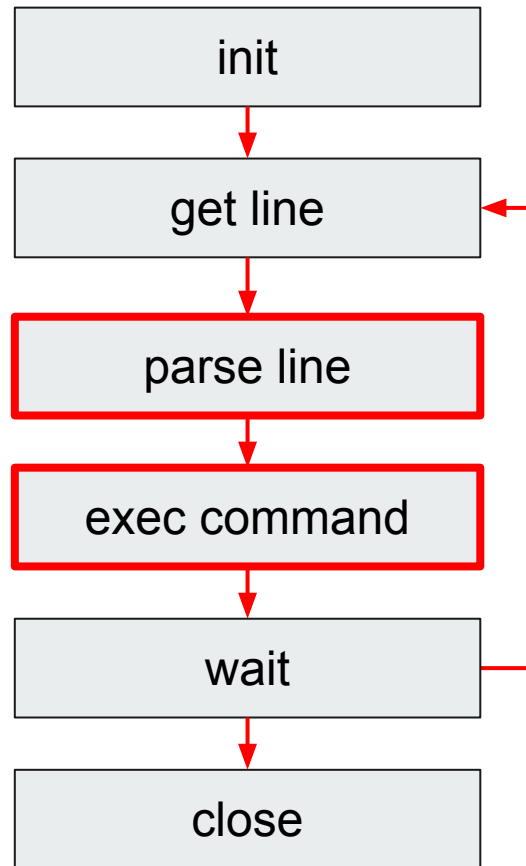
## Parte 2: Redirecciones (flujo estándar)

### Prueba 1:

1. `fd = open("file-out.txt")`
2. `close(1)`
3. `dup(fd)` // debería devolver 1
4. `close(fd)`

O también:

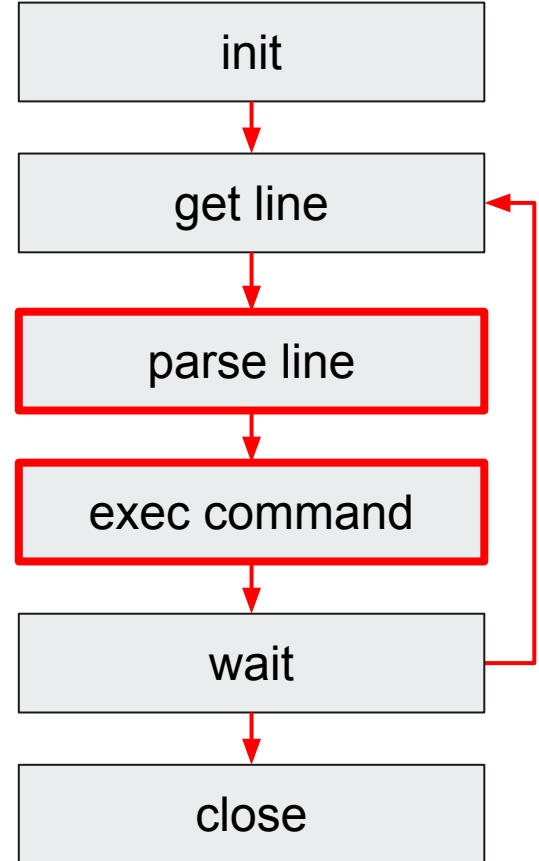
1. `fd = open("file-in.txt")`
2. `close(0)`
3. `dup(fd)` // debería devolver 0



## Parte 2: Redirecciones (flujo estándar)



```
int dup2(int oldfd, int newfd)
```

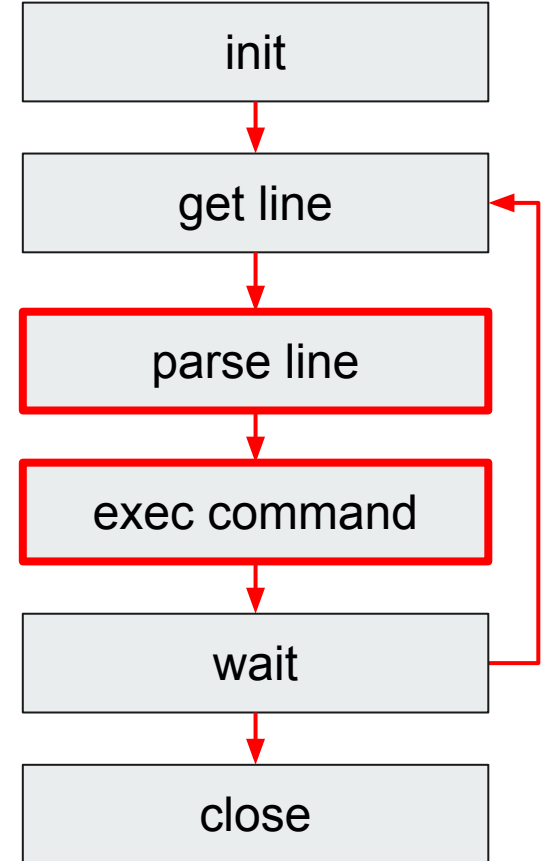


## Parte 2: Redirecciones (flujo estándar)

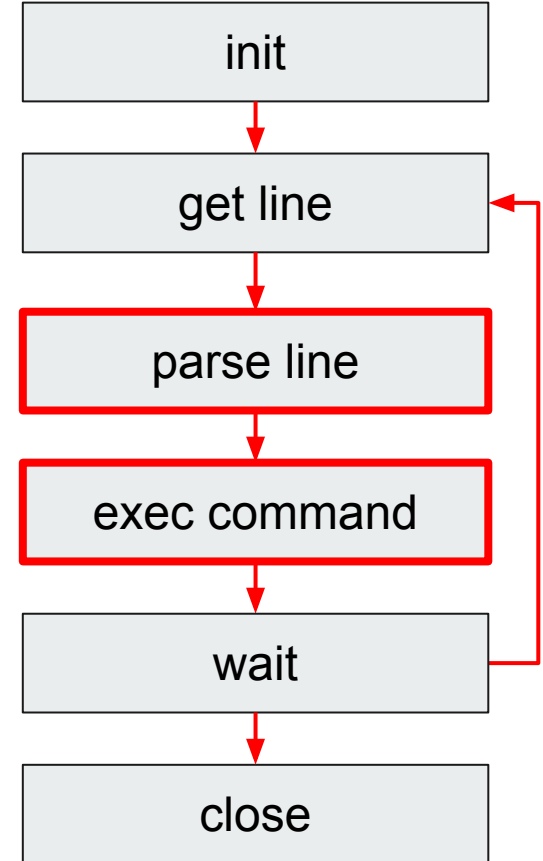
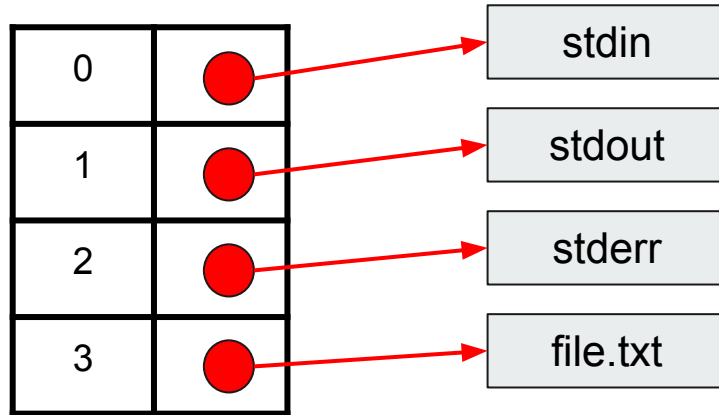
`int dup2(int oldfd, int newfd)`

Donde se quiere  
redireccionar el flujo  
estándar

El descriptor del flujo  
estándar que se quiere  
redireccionar

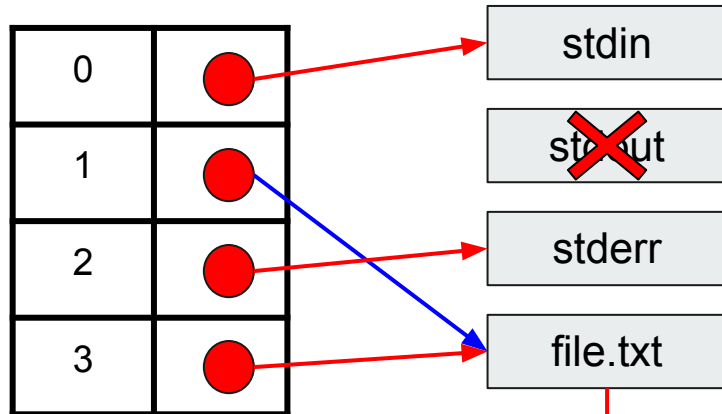


## Parte 2: Redirecciones (flujo estándar)

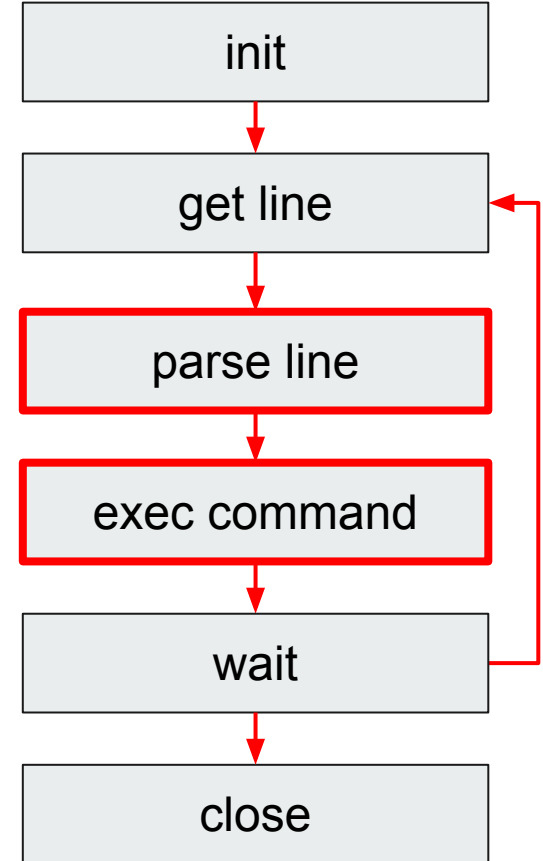


## Parte 2: Redirecciones (flujo estándar)

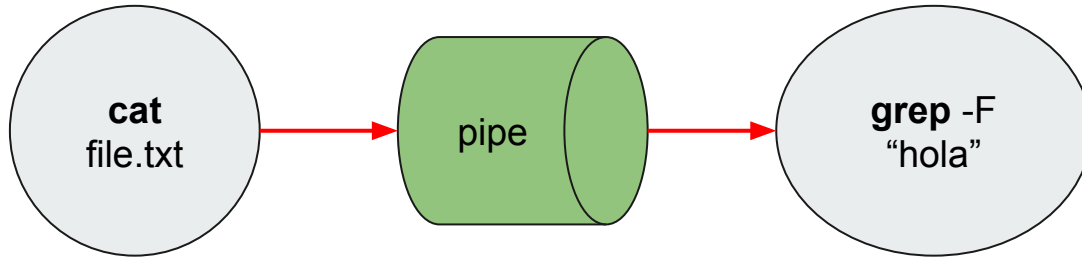
```
int dup2(3 // file.txt, 1 // stdout)
```



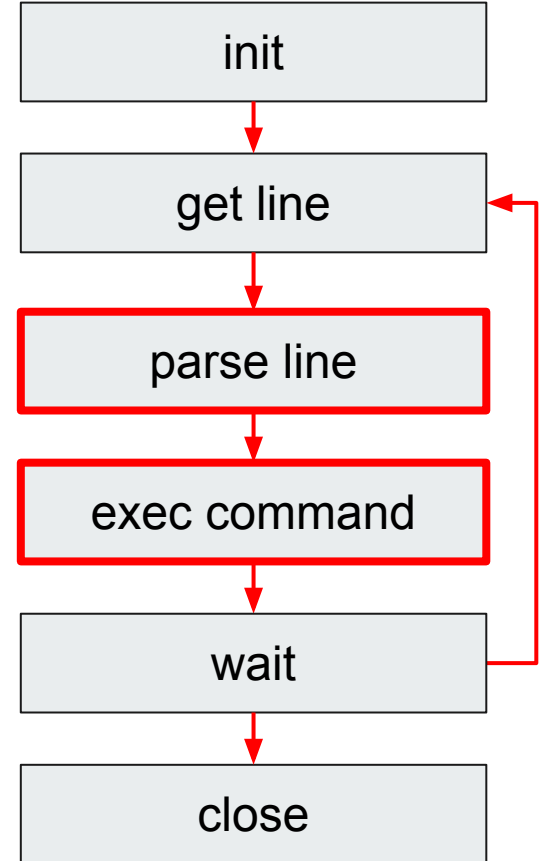
Tip: flag `O_CLOEXEC` de `open(2)`



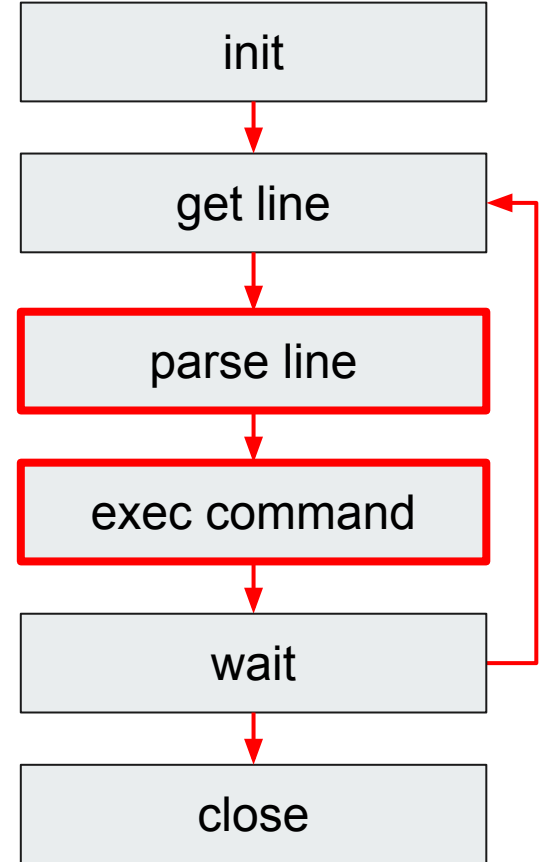
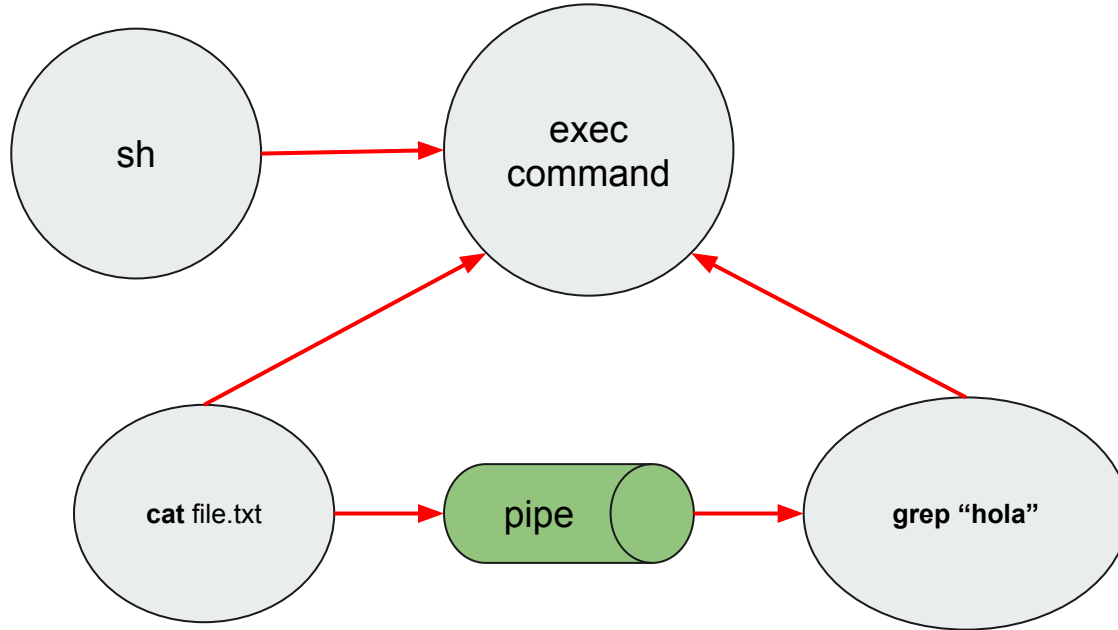
## Parte 2: Redirecciones (pipes)



*"tuberías simples"*

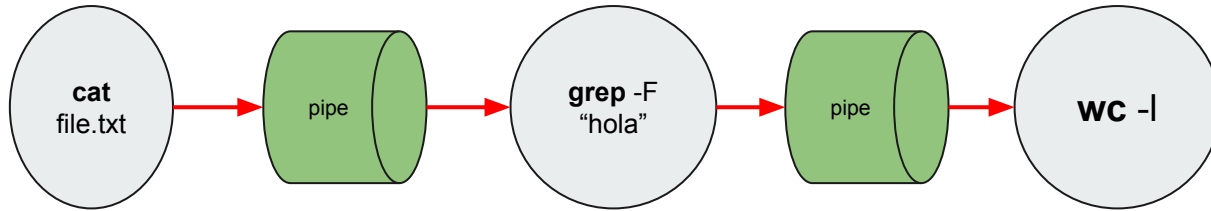


## Parte 2: Redirecciones (pipes)

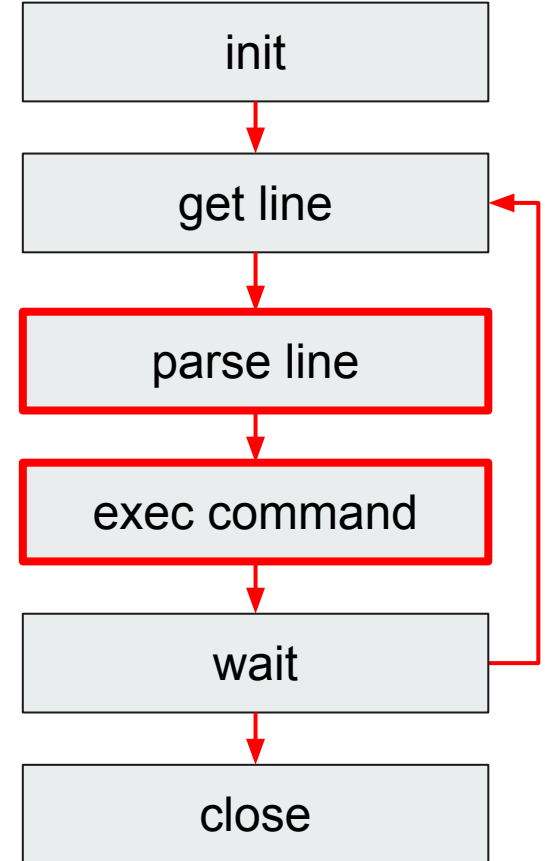




## Parte 2: Redirecciones (pipes)

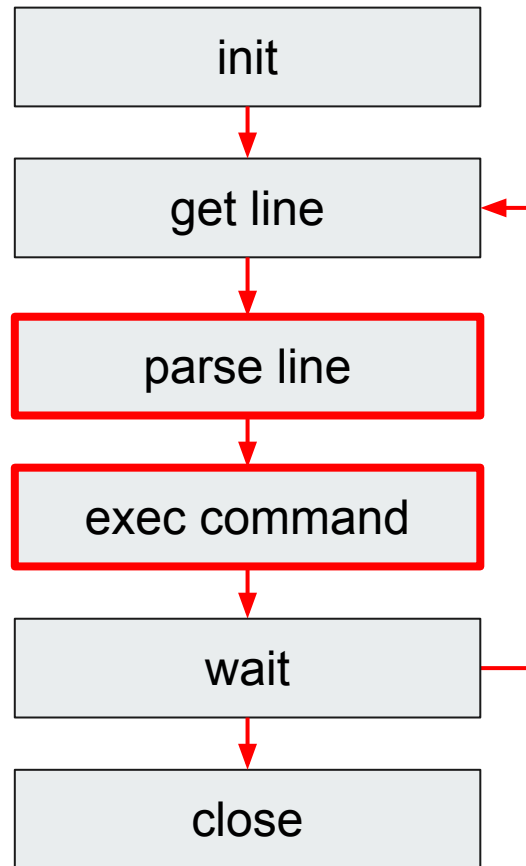
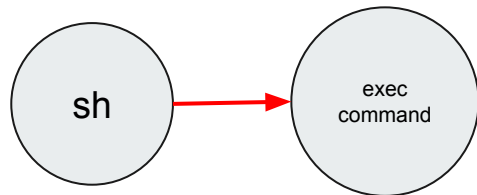


"tuberías múltiples"



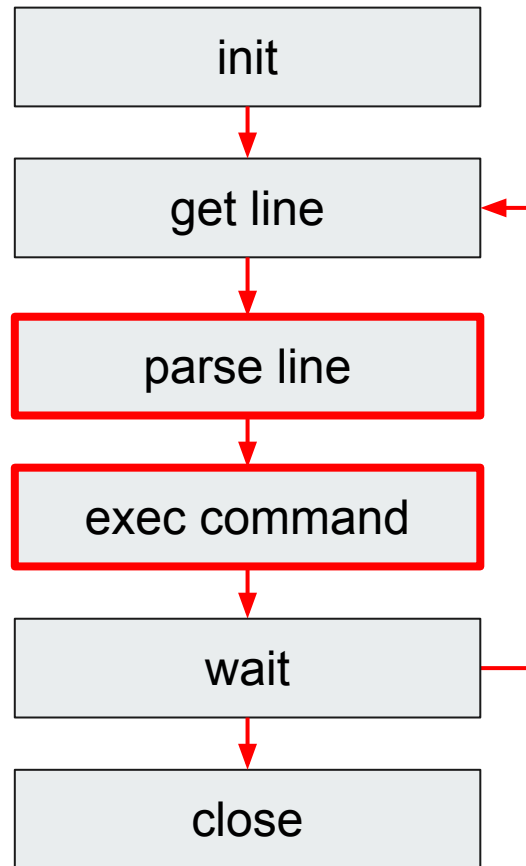
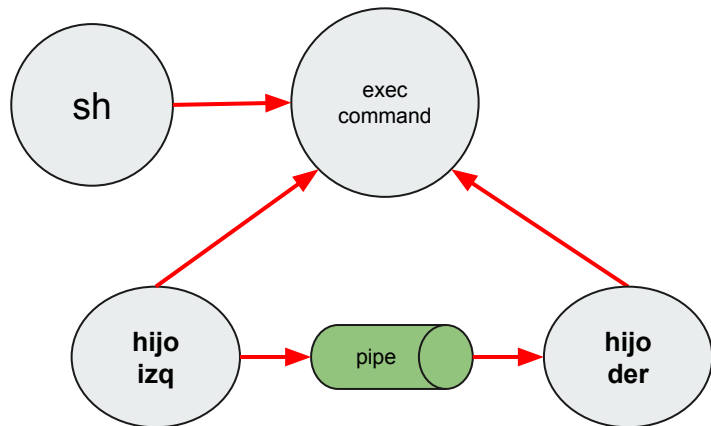
```
cat file.txt | grep -F hola | wc -l
```

## Parte 2: Redirecciones (pipes)



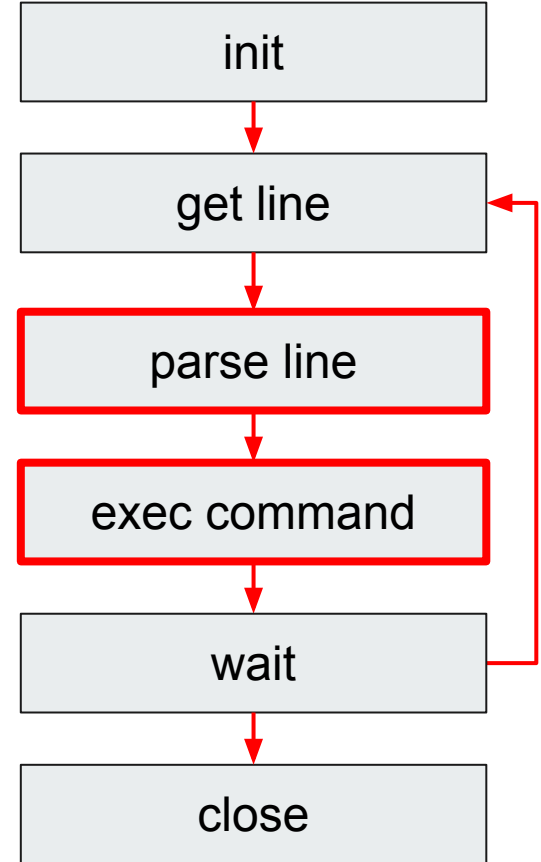
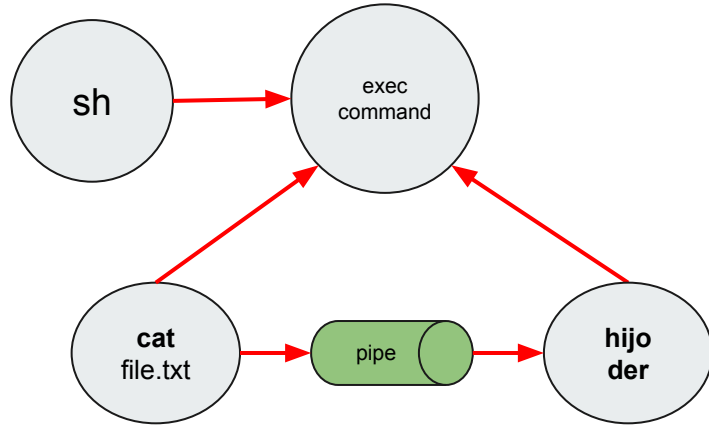
```
cat file.txt | grep -F hola | wc -l
```

## Parte 2: Redirecciones (pipes)



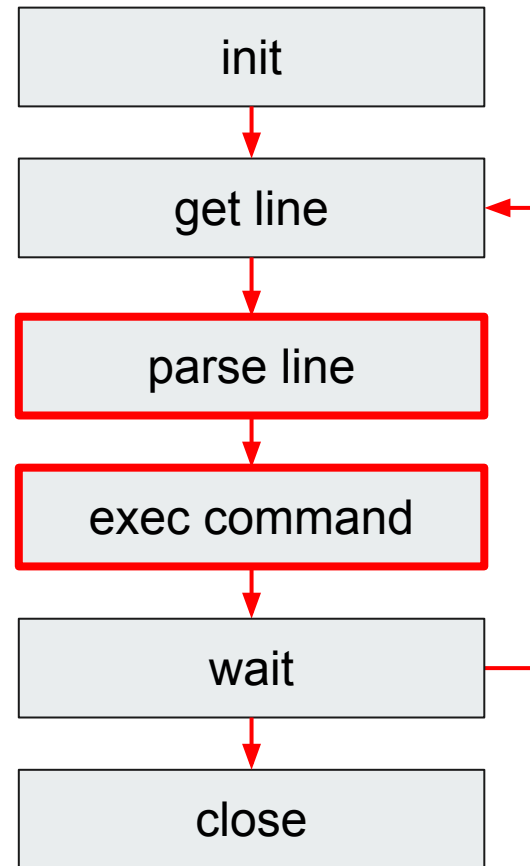
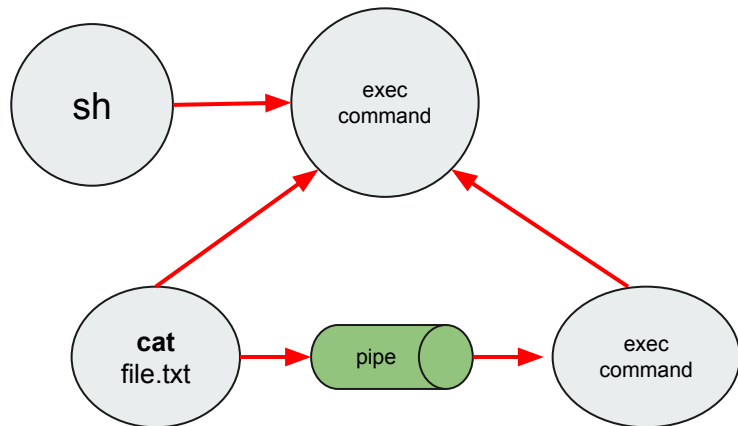
```
cat file.txt | grep -F hola | wc -l
```

## Parte 2: Redirecciones (pipes)



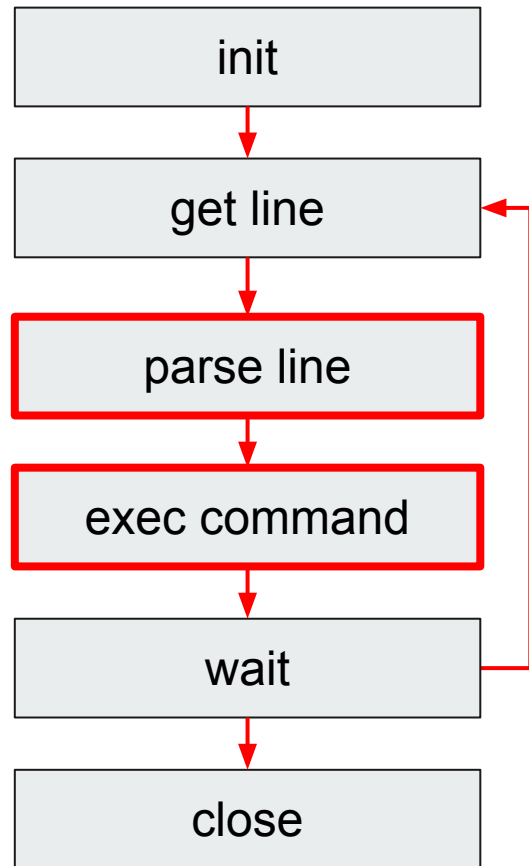
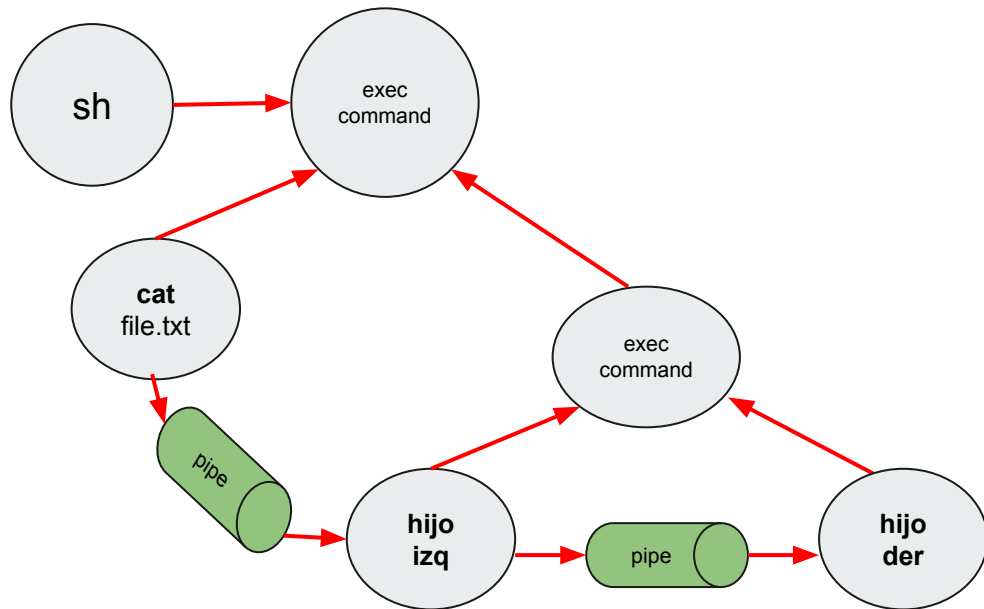
```
cat file.txt | grep -F hola | wc -l
```

## Parte 2: Redirecciones (pipes)



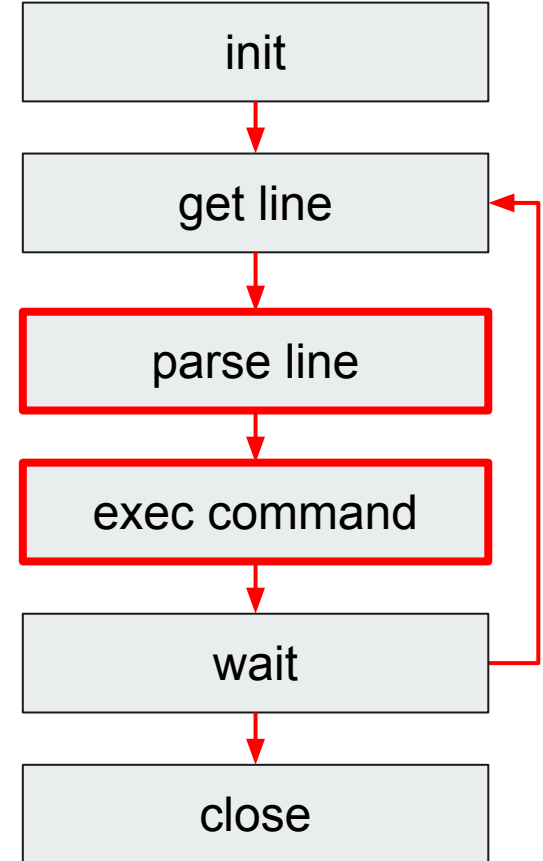
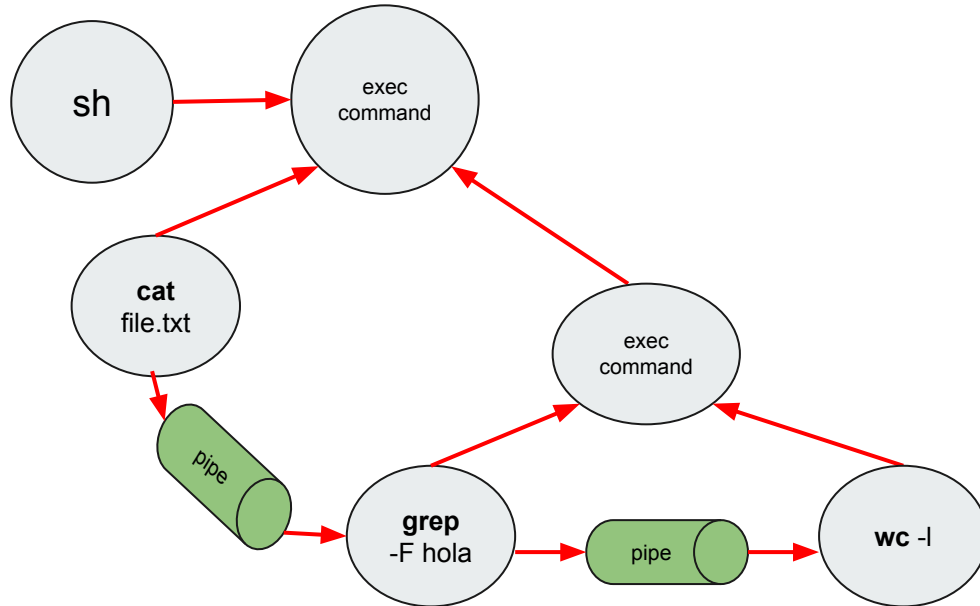
```
cat file.txt | grep -F hola | wc -l
```

## Parte 2: Redirecciones (pipes)



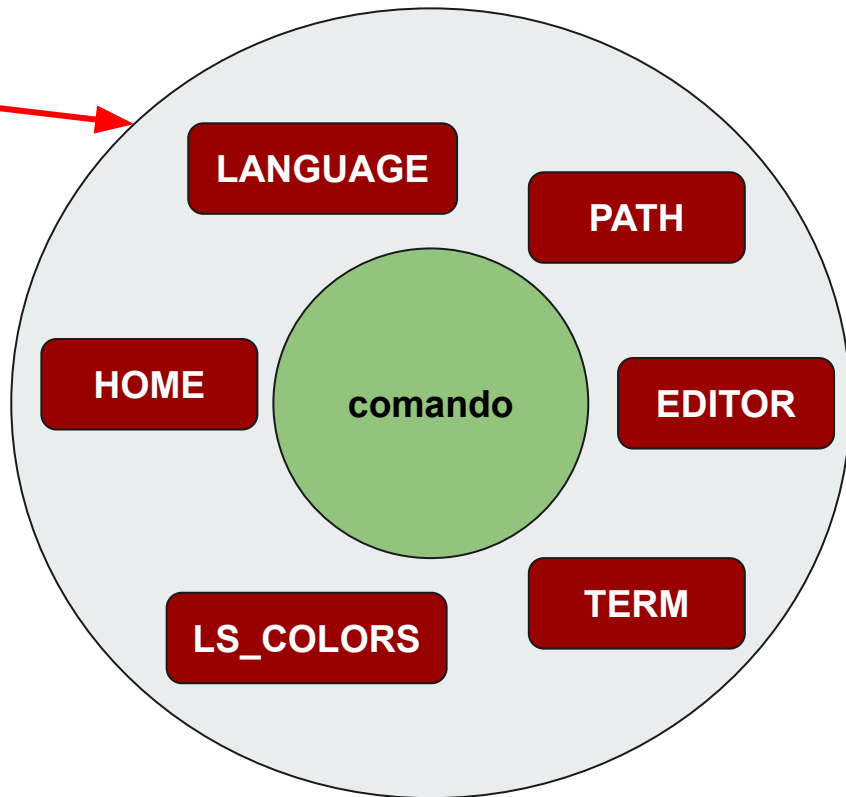
```
cat file.txt | grep -F hola | wc -l
```

## Parte 2: Redirecciones (pipes)



### Parte 3: Variables de entorno

```
$ echo $PATH
```

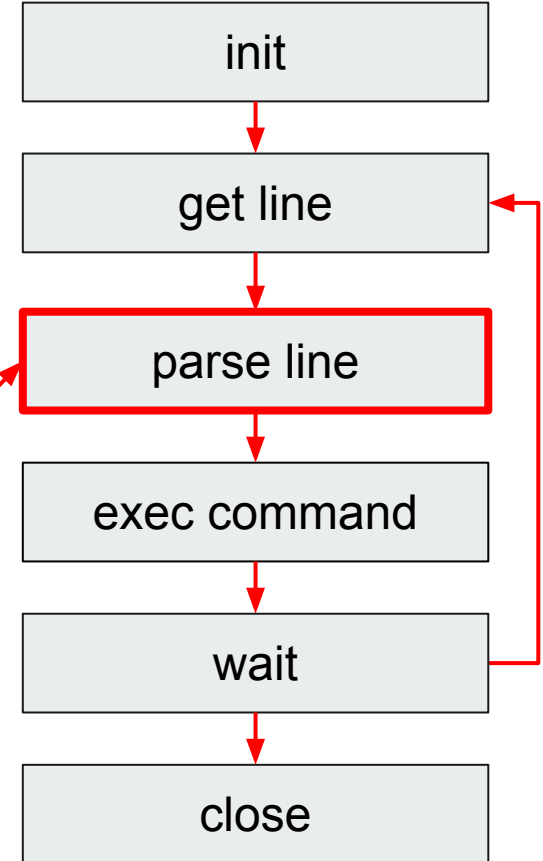




## Parte 3: Variables de entorno

```
$ echo $PATH
```

La “**expansión**” de estas *variables* ocurre antes que se ejecute el comando



## Parte 3: Variables de entorno

- *Ejemplo 1:*

```
$ echo $PATH
```

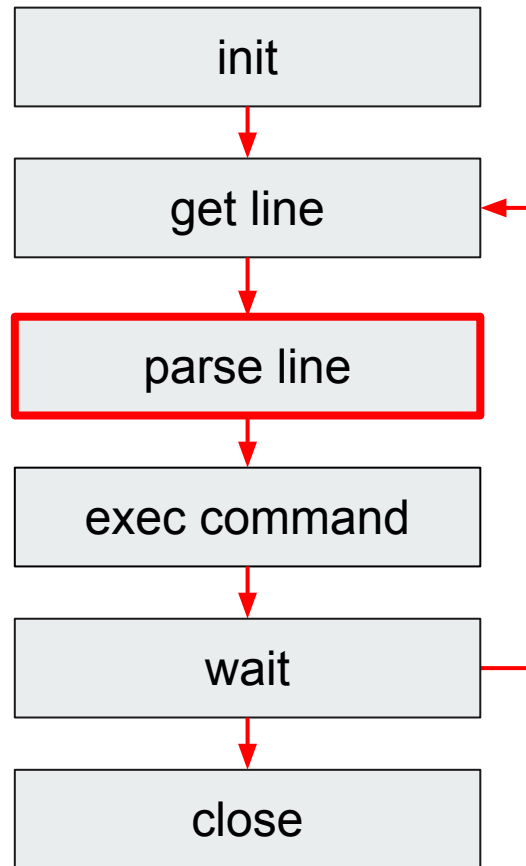
- *Ejemplo 2:*

```
$ echo $NO_EXISTE
```

- *Ejemplo 3:*

```
$ export NO_EXISTE=hola
```

```
$ echo $NO_EXISTE
```

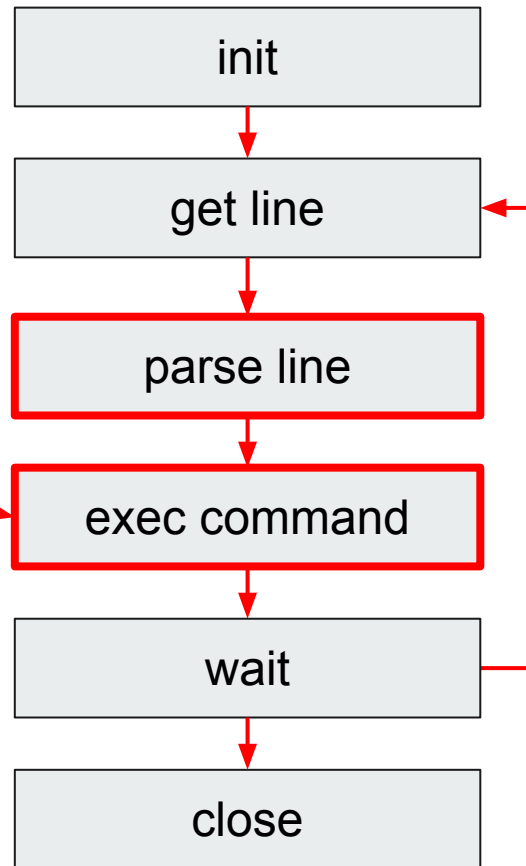


### Parte 3: Variables de entorno

Cómo hacemos si queremos agregar *variables de entorno* **temporales**?

```
$ REPO_ID=1234 TOKEN=my_token env
```

```
$ echo $REPO_ID $TOKEN
```



### Parte 3: Variables de entorno (mágicas)

Ejemplos:

```
$ echo $_
```

```
$ echo $$
```

```
$ echo $?
```

No existen en el ambiente/entorno del proceso. Son propias de la *shell*

*status code* del último proceso

## Parte 4: Extras (comandos *built-in*)

- **realizar acciones** en el proceso de la *shell*
- **optimizar** comandos

**\$ cd** /home/labs/shell

**\$ exit**

**\$ pwd**

## ¿Dónde aprender más?

- Páginas de manual (man)
- *The Linux Programming Interface*  
de Michael Kerrisk
- *Advanced Programming in the UNIX Environment*  
de Stevens & Rago

