

# Prácticas de Ampliación de Sistemas Operativos •

## Tarea Semana 2

---

### Descripción del programa `opening`

Escribe un programa llamado `opening` que acepte las siguientes opciones `opening -v VISOR [IMGs]`. Por ejemplo:

```
$ ./opening -v inkscape img1.png img2.svg img2.jpg
```

- Si no se especifica `-v VISOR`, se devolverá `EXIT_FAILURE` tras mostrar el siguiente mensaje: **Uso:** `./opening -v VISOR [IMGs]` en la salida estándar de error.
- Si el binario `VISOR` no se encuentra, se devolverá `EXIT_FAILURE` después de imprimir el siguiente mensaje: **Error: 'VISOR' no encontrado** en la salida estándar de error. *Pista:* Si intentamos ejecutar una llamada al sistema `exec*( )` y el binario no se puede encontrar, `exec*( )` fallará.
- Se debe crear un proceso hijo con la aplicación externa `VISOR` para abrir cada imagen `IMG`. Si no se especifica al menos una imagen `IMG`, se debe mostrar el siguiente mensaje **Error: No hay imágenes que visualizar** en la salida estándar de error y devolver `EXIT_FAILURE`. **Todos los procesos hijo deben crearse antes de esperar al primero.**
- La ejecución no debe concluir hasta que no se hayan cerrado todos los procesos hijo. **El proceso padre debe esperar a que concluyan todos ellos en orden de creación.** Por ejemplo, si el usuario teclea `./opening -v VISOR img1 img2 img3`, el proceso padre tiene que esperar primero al proceso hijo de `img1`, luego al de `img2` y por último al de `img3`.
- Cada vez que termine un proceso hijo, el proceso padre debe mostrar el mensaje `./opening: IMG: imgN.xyz, STATUS: status` donde `imgN.xyz` es el nombre del archivo del proceso hijo y `STATUS` su código de retorno.

### Ejemplos de ejecución de `opening`

```
$ ./opening
Uso: ./opening -v VISOR [IMGs]
$ echo $?
1

$ ./opening -v novisor
Error: No hay imágenes que visualizar
$ echo $?
1

$ ./opening -v inkscape
Error: No hay imágenes que visualizar
$ echo $?
1
```

```
$ ./openimg -v inkscape img1.png img2.svg img2.jpg
./openimg: IMG: img1.png, STATUS: 0
./openimg: IMG: img2.svg, STATUS: 0
./openimg: IMG: img2.jpg, STATUS: 0
$ echo $?
0

$ ./openimg -v inkscape img1.png NOIMG img2.jpg
./openimg: IMG: img1.png, STATUS: 0
./openimg: IMG: NOIMG, STATUS: 1
./openimg: IMG: img2.jpg, STATUS: 0
$ echo $?
0
```

## Entrega

La entrega consistirá en un único archivo `openimg.c`. El archivo debe compilar conforme a los archivos `tasks.json` o `Makefile` incluidos en `AS0-Semana1.tar.xz`, es decir, con `gcc -ggdb3 -Wall -Werror -Wno-unused -std=c11 openimg.c -o openimg`. La primera línea de `openimg.c` debe ser la directiva para compilar conforme al estándar POSIX: `#define _POSIX_C_SOURCE 200809L`.