

Tareas

Prácticas de Ampliación de Sistemas Operativos - Tarea Semana 2 - Corregida

Borrador - En proceso

Enviada

Corregida

Título	Prácticas de Ampliación de Sistemas Operativos - Tarea Semana 2
Estudiante	ELENA PEREZ GONZALEZ-TABLAS
Fecha de envío	12-oct-2020 16:01
Calificación	9,00 (máx 10,00)

Instrucciones

Descripción del programa openurl

Escribe un programa llamado `openurl` que acepte las siguientes opciones `openurl -n NAVEGADOR [URLs]`. Por ejemplo:

```
$ ./openurl -n firefox https://www.cruzroja.es https://www.msf.es  
https://www.unicef.org
```

- Si no se especifica `-n NAVEGADOR`, se devolverá `EXIT_FAILURE` tras mostrar el siguiente mensaje: `Uso: ./openurl -n NAVEGADOR [URLs]` en la salida estándar de error.
- Si el binario `NAVEGADOR` no se encuentra, el proceso hijo devolverá `EXIT_FAILURE` después de imprimir el siguiente mensaje: `Error: NAVEGADOR 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 `NAVEGADOR` para abrir cada URL. Si no se especifica al menos una URL, se abrirá la dirección `www.um.es`. 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 hijos. El proceso padre debe esperar a que concluyan todos ellos en orden de creación. Por ejemplo, si el usuario tecleó `./openurl -n navegador url1 url2 url3`, el proceso padre tiene que esperar primero al proceso hijo de `url1`, luego al de `url2` y, por último, al de `url3`.
- Cada vez que termine un proceso hijo, el proceso padre debe mostrar el mensaje `./openurl: URL: url1, STATUS: status` donde URL es la URL del proceso hijo y STATUS su código de retorno.

Ejemplos de ejecución de openurl

```
$ ./openurl
```

```
Uso: ./openurl -n NAVEGADOR [URLs]
```

```
$ ./openurl -n nonavegador
```

```
Error: nonavegador no encontrado
```

```
./openurl: URL: https://www.um.es, STATUS: 1
```

```
$ ./openurl -n firefox
```

```
./openurl: URL: https://www.um.es, STATUS: 0
```

```
$ ./openurl -n firefox https://www.cruzroja.es https://www.msf.es  
https://www.unicef.org
```

```
./openurl: URL: https://www.cruzroja.es, STATUS: 0
```

```
./openurl: URL: https://www.msf.es, STATUS: 0
```

```
./openurl: URL: https://www.unicef.org, STATUS: 0
```

Notas sobre los navegadores Chrome y Firefox

- En el navegador Firefox, para hacer que cada URL se abra en una ventana diferente, se tiene que incluir la opción `--new-window`.
- En el navegador Chrome, para hacer que cada URL se abra en una ventana diferente, se tiene que incluir la opción `--new-window`.

Entrega

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

Recursos adicionales para la tarea

No hay adjuntos todavía

Adjunto enviado

-  [openurl.c](#) (3 KB; 12-oct-2020 16:00)

Comentarios adicionales del profesor

El código es un poco confuso porque el padre también ejecuta las llamadas a hijos(). Sería mejor evitarlo poniendo la comprobación fuera.

[Volver a la lista](#)