

SOPER Practica 2

Generated by Doxygen 1.8.11

Contents

| | | |
|----------|--|----------|
| 1 | File Index | 1 |
| 1.1 | File List | 1 |
| 2 | File Documentation | 3 |
| 2.1 | ejercicio2.c File Reference | 3 |
| 2.1.1 | Detailed Description | 4 |
| 2.1.2 | Macro Definition Documentation | 4 |
| 2.1.2.1 | N_CHILDS | 4 |
| 2.2 | ejercicio4.c File Reference | 4 |
| 2.2.1 | Detailed Description | 5 |
| 2.2.2 | Macro Definition Documentation | 5 |
| 2.2.2.1 | N_PRINT | 5 |
| 2.2.3 | Function Documentation | 5 |
| 2.2.3.1 | handler_SIGUSR1(int sig) | 5 |
| 2.2.3.2 | is_valid_integer(char *input) | 6 |
| 2.3 | ejercicio6a.c File Reference | 7 |
| 2.3.1 | Detailed Description | 7 |
| 2.3.2 | Macro Definition Documentation | 8 |
| 2.3.2.1 | NUM | 8 |
| 2.4 | ejercicio6b.c File Reference | 8 |
| 2.4.1 | Detailed Description | 8 |
| 2.4.2 | Macro Definition Documentation | 9 |
| 2.4.2.1 | NUM | 9 |
| 2.4.3 | Function Documentation | 9 |

| | | |
|---------|--|-----------|
| 2.4.3.1 | handler_SIGTERM(int sig) | 9 |
| 2.5 | semaforos.c File Reference | 9 |
| 2.5.1 | Detailed Description | 10 |
| 2.5.2 | Function Documentation | 10 |
| 2.5.2.1 | Borrar_Semaforo(int semid) | 10 |
| 2.5.2.2 | Crear_Semaforo(key_t key, int size, int *semid) | 11 |
| 2.5.2.3 | Down_Semaforo(int id, int num_sem, int undo) | 11 |
| 2.5.2.4 | DownMultiple_Semaforo(int id, int size, int undo, int *active) | 11 |
| 2.5.2.5 | Inicializar_Semaforo(int semid, unsigned short *array) | 11 |
| 2.5.2.6 | Up_Semaforo(int id, int num_sem, int undo) | 12 |
| 2.5.2.7 | UpMultiple_Semaforo(int id, int size, int undo, int *active) | 12 |
| 2.6 | semaforos.h File Reference | 12 |
| 2.6.1 | Detailed Description | 14 |
| 2.6.2 | Macro Definition Documentation | 14 |
| 2.6.2.1 | ERROR | 14 |
| 2.6.2.2 | OK | 14 |
| 2.6.3 | Function Documentation | 14 |
| 2.6.3.1 | Borrar_Semaforo(int semid) | 14 |
| 2.6.3.2 | Crear_Semaforo(key_t key, int size, int *semid) | 15 |
| 2.6.3.3 | Down_Semaforo(int id, int num_sem, int undo) | 15 |
| 2.6.3.4 | DownMultiple_Semaforo(int id, int size, int undo, int *active) | 15 |
| 2.6.3.5 | Inicializar_Semaforo(int semid, unsigned short *array) | 16 |
| 2.6.3.6 | Up_Semaforo(int id, int num_sem, int undo) | 16 |
| 2.6.3.7 | UpMultiple_Semaforo(int id, int size, int undo, int *active) | 16 |
| | Index | 17 |

Chapter 1

File Index

1.1 File List

Here is a list of all documented files with brief descriptions:

| | | |
|-------------------------------|--|----|
| ejercicio2.c | Ejercicio2 Creación de 4 procesos hijos que duermen durante 30s mientras que el proceso padre espera 5s y después envía a cada proceso hijo la señal de finalización SIGTERM | 3 |
| ejercicio4.c | Ejercicio4 Manejo de señales entre procesos hijos y padre | 4 |
| ejercicio6a.c | Ejercicio6a Toma de contacto con la funcion alarm() | 7 |
| ejercicio6b.c | Ejercicio6b Toma de contacto con la funcion sigfillset() | 8 |
| semaforos.c | Ejercicio9 | 9 |
| semaforos.h | Utilidades de manejo de semaforos | 12 |

Chapter 2

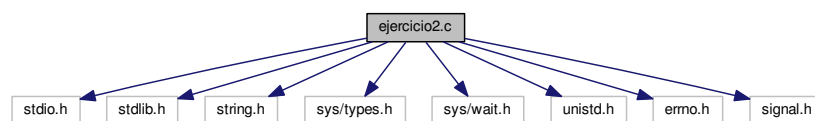
File Documentation

2.1 ejercicio2.c File Reference

Ejercicio2 Creación de 4 procesos hijos que duermen durante 30s mientras que el proceso padre espera 5s y después envía a cada proceso hijo la señal de finalización SIGTERM.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
```

Include dependency graph for ejercicio2.c:



Macros

- `#define N_CHILDEN 4`

Functions

- `int main ()`

2.1.1 Detailed Description

Ejercicio2 Creación de 4 procesos hijos que duermen durante 30s mientras que el proceso padre espera 5s y después envía a cada proceso hijo la señal de finalización SIGTERM.

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

31-03-2018

2.1.2 Macro Definition Documentation

2.1.2.1 #define N_CHILDS 4

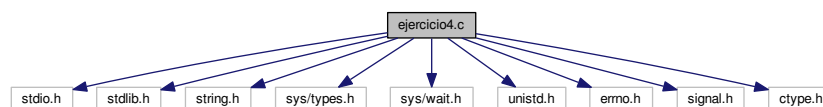
Número de procesos hijos

2.2 ejercicio4.c File Reference

Ejercicio4 Manejo de señales entre procesos hijos y padre.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <ctype.h>
```

Include dependency graph for ejercicio4.c:



Macros

- #define `N_PRINT` 10

Functions

- int `is_valid_integer` (char *input)
evalua si un argumento de entrada es verdaderamente un entero
- void `handler_SIGUSR1` (int sig)
nuevo manejador de la señal SIGUSR1. Su funcionalidad es simplemente no hacer nada.
- int `main` (int argc, char *argv[])

2.2.1 Detailed Description

Ejercicio4 Manejo de señales entre procesos hijos y padre.

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

31-03-2018

2.2.2 Macro Definition Documentation

2.2.2.1 #define N_PRINT 10

Número de impresiones por pantalla de cada hijo

2.2.3 Function Documentation

2.2.3.1 void handler_SIGUSR1 (int sig)

nuevo manejador de la señal SIGUSR1. Su funcionalidad es simplemente no hacer nada.

Parameters

| | |
|------------|--|
| <i>sig</i> | - la señal que utilizará este manejador cuando sea llamada |
|------------|--|

Returns

void.

2.2.3.2 `int is_valid_integer (char * input)`

evalua si un argumento de entrada es verdaderamente un entero

Parameters

| | |
|--------------|--|
| <i>input</i> | - contiene la cadena de caracteres sospechosa de ser un entero |
|--------------|--|

Returns

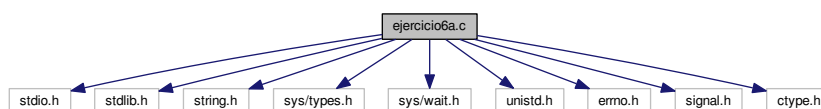
1 si es un entero, 0 en caso contrario.

2.3 ejercicio6a.c File Reference

Ejercicio6a Toma de contacto con la funcion alarm()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <ctype.h>
```

Include dependency graph for ejercicio6a.c:



Macros

- `#define NUM 5`

Functions

- `int main (int argc, char *argv[])`

2.3.1 Detailed Description

Ejercicio6a Toma de contacto con la funcion alarm()

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

31-03-2018

2.3.2 Macro Definition Documentation

2.3.2.1 #define NUM 5

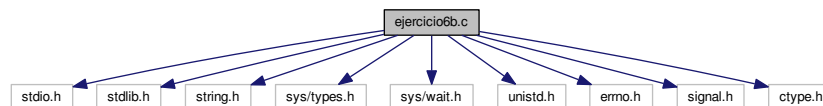
Número de impresiones

2.4 ejercicio6b.c File Reference

Ejercicio6b Toma de contacto con la funcion sigfillset()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <ctype.h>
```

Include dependency graph for ejercicio6b.c:



Macros

- #define NUM 5

Functions

- void [handler_SIGTERM](#) (int sig)
nuevo manejador de la señal SIGTERM. Su funcionalidad es simplemente imprimir el PID y un mensaje.
- int **main** (int argc, char *argv[])

2.4.1 Detailed Description

Ejercicio6b Toma de contacto con la funcion sigfillset()

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

31-03-2018

2.4.2 Macro Definition Documentation

2.4.2.1 #define NUM 5

Número de impresiones

2.4.3 Function Documentation

2.4.3.1 void handler_SIGTERM (int sig)

nuevo manejador de la señal SIGTERM. Su funcionalidad es simplemente imprimir el PID y un mensaje.

Parameters

| | |
|-----|--|
| sig | - la señal que utilizará este manejador cuando sea llamada |
|-----|--|

Returns

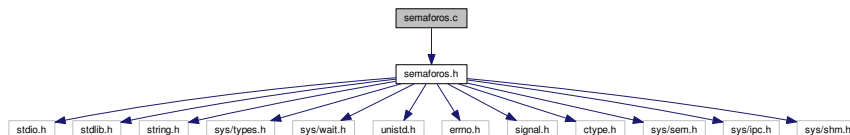
void.

2.5 semaforos.c File Reference

Ejercicio9.

```
#include "semaforos.h"
```

Include dependency graph for semaforos.c:



Functions

- int [Inicializar_Semaforo](#) (int semid, unsigned short *array)
funcion inicializadora de semaforos. Su funcionalidad es inicializar los semaforos indicados.
- int [Borrar_Semaforo](#) (int semid)
funcion que borra semaforos. Su funcionalidad es borrar el semaforo indicado.
- int [Crear_Semaforo](#) (key_t key, int size, int *semid)
funcion creadora de semaforos. Su funcionalidad es crear un semaforo con la clave y tamaño indicados.
- int [Down_Semaforo](#) (int id, int num_sem, int undo)
funcion que decrementa un semaforo. Su funcionalidad es bajar o decrementar el semaforo indicado.
- int [DownMultiple_Semaforo](#) (int id, int size, int undo, int *active)
funcion que decrementa varios semaforos. Su funcionalidad es bajar o decrementar todos los semaforos indicados.
- int [Up_Semaforo](#) (int id, int num_sem, int undo)
funcion que incrementa un semaforo. Su funcionalidad es subir o incrementar el semaforo indicado.
- int [UpMultiple_Semaforo](#) (int id, int size, int undo, int *active)
funcion que incrementa varios semaforos. Su funcionalidad es subir o incrementar todos los semaforos indicados.

2.5.1 Detailed Description

Ejercicio9.

Utilidades de manejo de semaforos.

Este ejercicio comprueba la librería de semaforos mientras ayuda didacticamente

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

06-04-2018

Este modulo contiene la implementacion de las funciones de manejo de semaforos.

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

01-04-2018

2.5.2 Function Documentation

2.5.2.1 int Borrar_Semaforo (int *semid*)

funcion que borra semaforos. Su funcionalidad es borrar el semaforo indicado.

Parameters

| | |
|--------------|------------------------------|
| <i>semid</i> | - identificador del semaforo |
|--------------|------------------------------|

Returns

OK si todo fue correcto, ERROR en caso de error.

2.5.2.2 int Crear_Semaforo (key_t key, int size, int * semid)

funcion creadora de semaforos. Su funcionalidad es crear un semaforo con la clave y tamaño indicados.

Parameters

| | |
|--------------|---|
| <i>key</i> | - clave precompartida del semaforo |
| <i>size</i> | - tamaño del semaforo. |
| <i>semid</i> | - entero pasado por referencia para alojar el semid del semaforo creado |

Returns

int *semid - identificador del semaforo. Int ERROR en caso de error, 1 si el semaforo ya existia, 0 en caso de exito.

2.5.2.3 int Down_Semaforo (int id, int num_sem, int undo)

funcion que decrementa un semaforo. Su funcionalidad es bajar o decrementar el semaforo indicado.

Parameters

| | |
|----------------|--|
| <i>id</i> | - identificador del semaforo |
| <i>num_sem</i> | - semaforo dentro del array |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.5.2.4 int DownMultiple_Semaforo (int id, int size, int undo, int * active)

funcion que decrementa varios semaforos. Su funcionalidad es bajar o decrementar todos los semaforos indicados.

Parameters

| | |
|---------------|---|
| <i>id</i> | - identificador del semaforo |
| <i>size</i> | - tamaño del array de ids de los semaforos involucrados(active) |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |
| <i>active</i> | Semaforos involucrados. |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.5.2.5 int Inicializar_Semaforo (int semid, unsigned short * array)

funcion inicializadora de semaforos. Su funcionalidad es inicializar los semaforos indicados.

Parameters

| | |
|--------------|------------------------------|
| <i>semid</i> | - identificador del semaforo |
| <i>array</i> | - valores iniciales |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.5.2.6 int Up_Semaforo (int *id*, int *num_sem*, int *undo*)

funcion que incrementa un semaforo. Su funcionalidad es subir o incrementar el semaforo indicado.

Parameters

| | |
|----------------|--|
| <i>id</i> | - identificador del semaforo |
| <i>num_sem</i> | - semaforo dentro del array |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.5.2.7 int UpMultiple_Semaforo (int *id*, int *size*, int *undo*, int * *active*)

funcion que incrementa varios semaforos. Su funcionalidad es subir o incrementar todos los semaforos indicados.

Parameters

| | |
|---------------|---|
| <i>id</i> | - identificador del semaforo |
| <i>size</i> | - tamaño del array de ids de los semaforos involucrados(active) |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |
| <i>active</i> | Semaforos involucrados. |

Returns

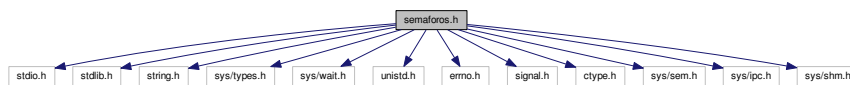
OK si todo fue correcto, ERROR en caso de error.

2.6 semaforos.h File Reference

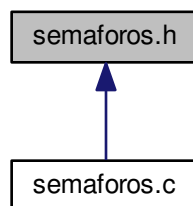
Utilidades de manejo de semaforos.


```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <ctype.h>
#include <sys/sem.h>
#include <sys/ipc.h>
#include <sys/shm.h>
```

Include dependency graph for semaforos.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define ERROR -1`
- `#define OK 0`

Functions

- `int Inicializar_Semaforo (int semid, unsigned short *array)`
funcion inicializadora de semaforos. Su funcionalidad es inicializar los semaforos indicados.
- `int Borrar_Semaforo (int semid)`
funcion que borra semaforos. Su funcionalidad es borrar el semaforo indicado.
- `int Crear_Semaforo (key_t key, int size, int *semid)`
funcion creadora de semaforos. Su funcionalidad es crear un semaforo con la clave y tamaño indicados.
- `int Down_Semaforo (int id, int num_sem, int undo)`
funcion que decrementa un semaforo. Su funcionalidad es bajar o decrementar el semaforo indicado.

- int [DownMultiple_Semaforo](#) (int id, int size, int undo, int *active)
funcion que decrementa varios semaforos. Su funcionalidad es bajar o decrementar todos los semaforos indicados.
- int [Up_Semaforo](#) (int id, int num_sem, int undo)
funcion que incrementa un semaforo. Su funcionalidad es subir o incrementar el semaforo indicado.
- int [UpMultiple_Semaforo](#) (int id, int size, int undo, int *active)
funcion que incrementa varios semaforos. Su funcionalidad es subir o incrementar todos los semaforos indicados.

2.6.1 Detailed Description

Utilidades de manejo de semaforos.

Este modulo contiene los prototipos de las funciones de manejo de semaforos.

Author

Alejandro Santorum & David Cabornero (G2202-Pareja7)

Version

1.0

Date

01-04-2018

2.6.2 Macro Definition Documentation

2.6.2.1 #define ERROR -1

Constante con significado de operacion erronea

2.6.2.2 #define OK 0

Constante con significado de operacion exitosa

2.6.3 Function Documentation

2.6.3.1 int Borrar_Semaforo (int *semid*)

funcion que borra semaforos. Su funcionalidad es borrar el semaforo indicado.

Parameters

| | |
|--------------|------------------------------|
| <i>semid</i> | - identificador del semaforo |
|--------------|------------------------------|

Returns

OK si todo fue correcto, ERROR en caso de error.

2.6.3.2 int Crear_Semaforo (key_t key, int size, int * semid)

funcion creadora de semaforos. Su funcionalidad es crear un semaforo con la clave y tamaño indicados.

Parameters

| | |
|--------------|---|
| <i>key</i> | - clave precompartida del semaforo |
| <i>size</i> | - tamaño del semaforo. |
| <i>semid</i> | - entero pasado por referencia para alojar el semid del semaforo creado |

Returns

int *semid - identificador del semaforo. Int ERROR en caso de error, 1 si el semaforo ya existia, 0 en caso de exito.

2.6.3.3 int Down_Semaforo (int id, int num_sem, int undo)

funcion que decrementa un semaforo. Su funcionalidad es bajar o decrementar el semaforo indicado.

Parameters

| | |
|----------------|--|
| <i>id</i> | - identificador del semaforo |
| <i>num_sem</i> | - semaforo dentro del array |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.6.3.4 int DownMultiple_Semaforo (int id, int size, int undo, int * active)

funcion que decrementa varios semaforos. Su funcionalidad es bajar o decrementar todos los semaforos indicados.

Parameters

| | |
|---------------|---|
| <i>id</i> | - identificador del semaforo |
| <i>size</i> | - tamaño del array de ids de los semaforos involucrados(active) |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |
| <i>active</i> | Semaforos involucrados. |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.6.3.5 int Inicializar_Semaforo (int *semid*, unsigned short * *array*)

funcion inicializadora de semaforos. Su funcionalidad es inicializar los semaforos indicados.

Parameters

| | |
|--------------|------------------------------|
| <i>semid</i> | - identificador del semaforo |
| <i>array</i> | - valores iniciales |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.6.3.6 int Up_Semaforo (int *id*, int *num_sem*, int *undo*)

funcion que incrementa un semaforo. Su funcionalidad es subir o incrementar el semaforo indicado.

Parameters

| | |
|----------------|--|
| <i>id</i> | - identificador del semaforo |
| <i>num_sem</i> | - semaforo dentro del array |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |

Returns

OK si todo fue correcto, ERROR en caso de error.

2.6.3.7 int UpMultiple_Semaforo (int *id*, int *size*, int *undo*, int * *active*)

funcion que incrementa varios semaforos. Su funcionalidad es subir o incrementar todos los semaforos indicados.

Parameters

| | |
|---------------|---|
| <i>id</i> | - identificador del semaforo |
| <i>size</i> | - tamaño del array de ids de los semaforos involucrados(active) |
| <i>undo</i> | - flag de modo persistente pese a finalizacion abrupta |
| <i>active</i> | Semaforos involucrados. |

Returns

OK si todo fue correcto, ERROR en caso de error.

Index

Borrar_Semaforo
semaforos.c, [10](#)
semaforos.h, [14](#)

Crear_Semaforo
semaforos.c, [10](#)
semaforos.h, [15](#)

Down_Semaforo
semaforos.c, [11](#)
semaforos.h, [15](#)

DownMultiple_Semaforo
semaforos.c, [11](#)
semaforos.h, [15](#)

ERROR
semaforos.h, [14](#)

ejercicio2.c, [3](#)
N_CHILDS, [4](#)
ejercicio4.c, [4](#)
handler_SIGUSR1, [5](#)
is_valid_integer, [5](#)
N_PRINT, [5](#)
ejercicio6a.c, [7](#)
NUM, [8](#)
ejercicio6b.c, [8](#)
handler_SIGTERM, [9](#)
NUM, [9](#)

handler_SIGTERM
ejercicio6b.c, [9](#)
handler_SIGUSR1
ejercicio4.c, [5](#)

Inicializar_Semaforo
semaforos.c, [11](#)
semaforos.h, [16](#)
is_valid_integer
ejercicio4.c, [5](#)

N_CHILDS
ejercicio2.c, [4](#)

N_PRINT
ejercicio4.c, [5](#)

NUM
ejercicio6a.c, [8](#)
ejercicio6b.c, [9](#)

OK
semaforos.h, [14](#)

semaforos.c, [9](#)
Borrar_Semaforo, [10](#)
Crear_Semaforo, [10](#)
Down_Semaforo, [11](#)
DownMultiple_Semaforo, [11](#)
Inicializar_Semaforo, [11](#)
Up_Semaforo, [12](#)
UpMultiple_Semaforo, [12](#)
semaforos.h, [12](#)
Borrar_Semaforo, [14](#)
Crear_Semaforo, [15](#)
Down_Semaforo, [15](#)
DownMultiple_Semaforo, [15](#)
ERROR, [14](#)
Inicializar_Semaforo, [16](#)
OK, [14](#)
Up_Semaforo, [16](#)
UpMultiple_Semaforo, [16](#)

Up_Semaforo
semaforos.c, [12](#)
semaforos.h, [16](#)
UpMultiple_Semaforo
semaforos.c, [12](#)
semaforos.h, [16](#)