

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	MAX_HIJOS	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	NUM_ITER	5
2.2.3	Function Documentation	5
2.2.3.1	aredigits(char *string)	5
2.2.3.2	manejador_SIGUSR1(int sig)	6
2.3	ejercicio6a.c File Reference	6
2.3.1	Detailed Description	7
2.3.2	Macro Definition Documentation	7
2.3.2.1	NUM_PROC	7
2.3.2.2	SECS	7
2.4	ejercicio6b.c File Reference	7
2.4.1	Detailed Description	8
2.4.2	Macro Definition Documentation	8
2.4.2.1	NUM_PROC	8

2.4.2.2	SECS	8
2.4.3	Function Documentation	8
2.4.3.1	manejador_SIGTERM(int sig)	8
2.5	ejercicio9.c File Reference	9
2.5.1	Detailed Description	10
2.5.2	Macro Definition Documentation	10
2.5.2.1	DATDIR	10
2.5.2.2	ESCRIBE	10
2.5.2.3	KEY	10
2.5.2.4	LEE	10
2.5.2.5	MAXBUFF	11
2.5.2.6	NUM_CAJ	11
2.5.2.7	NUM_OPER	11
2.5.2.8	PATH	11
2.5.2.9	SIGDONE	11
2.5.2.10	SIGMONEY	11
2.5.2.11	TEXTDIR	11
2.5.3	Function Documentation	11
2.5.3.1	cajero(int id)	11
2.5.3.2	dinero_recogido(char *filename, bool dinero_sacado, int flag)	12
2.5.3.3	handle_SIGDONE(int sig, siginfo_t *info, void *vp)	12
2.5.3.4	handle_SIGMONEY(int sig, siginfo_t *info, void *vp)	12
2.5.3.5	increase_subtotal(char *filename, float delta)	12
2.5.3.6	randNum(float inf, float sup)	14
2.6	mysignal.c File Reference	14
2.6.1	Detailed Description	15
2.6.2	Function Documentation	15
2.6.2.1	sigaddset_var(sigset_t *sigt, int sig,...)	15
2.7	mysignal.h File Reference	15
2.7.1	Detailed Description	16

2.7.2	Function Documentation	16
2.7.2.1	sigaddset_var(sigset_t *sig, int sig,...)	16
2.8	semaforos.c File Reference	17
2.8.1	Detailed Description	18
2.8.2	Function Documentation	18
2.8.2.1	borrar_semaforo(int semid)	18
2.8.2.2	crear_semaforo(key_t key, int size, int *semid)	18
2.8.2.3	down_multiple_semaforo(int semid, int size, int undo, int *active)	19
2.8.2.4	down_semaforo(int semid, int num_sem, int undo)	19
2.8.2.5	inicializar_semaforo(int semid, unsigned short *array)	19
2.8.2.6	up_multiple_semaforo(int semid, int size, int undo, int *active)	19
2.8.2.7	up_semaforo(int semid, int num_sem, int undo)	20
2.9	semaforos.h File Reference	20
2.9.1	Detailed Description	21
2.9.2	Macro Definition Documentation	21
2.9.2.1	ERROR	21
2.9.2.2	OK	21
2.9.3	Function Documentation	21
2.9.3.1	borrar_semaforo(int semid)	21
2.9.3.2	crear_semaforo(key_t key, int size, int *semid)	22
2.9.3.3	down_multiple_semaforo(int semid, int size, int undo, int *active)	22
2.9.3.4	down_semaforo(int semid, int num_sem, int undo)	22
2.9.3.5	inicializar_semaforo(int semid, unsigned short *array)	23
2.9.3.6	up_multiple_semaforo(int semid, int size, int undo, int *active)	23
2.9.3.7	up_semaforo(int semid, int num_sem, int undo)	23
Index		25

Chapter 1

File Index

1.1 File List

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

ejercicio2.c	
Ejercicio 2	3
ejercicio4.c	
Ejercicio 4	4
ejercicio6a.c	
Ejercicio 6a	6
ejercicio6b.c	
Ejercicio 6b	7
ejercicio9.c	
Ejercicio 9	9
mysignal.c	
Funciones personales acerca de señales. Implementacion de mysignal.h	14
mysignal.h	
Interfaz de funciones personales acerca de señales	15
semaforos.c	
Funciones personales acerca de semaforos. Implementacion	17
semaforos.h	
Interfaz de funciones personales acerca de semaforos	20

Chapter 2

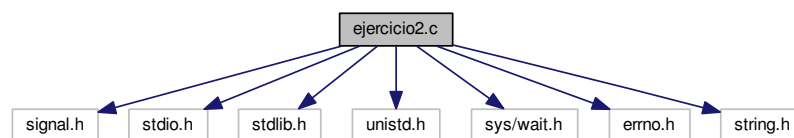
File Documentation

2.1 ejercicio2.c File Reference

Ejercicio 2.

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

Include dependency graph for ejercicio2.c:



Macros

- `#define MAX_HIJOS 4`

Functions

- `void main ()`

2.1.1 Detailed Description

Ejercicio 2.

Este fichero contiene el código fuente del ejercicio 2 de la entrega.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.1.2 Macro Definition Documentation

2.1.2.1 #define MAX_HIJOS 4

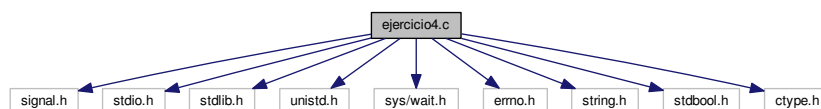
Numero maximo de hijos

2.2 ejercicio4.c File Reference

Ejercicio 4.

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

Include dependency graph for ejercicio4.c:



Macros

- #define NUM_ITER 10

Functions

- bool `aredigits` (char *string)
Comprueba si una string es numerica.
- void `manejador_SIGUSR1` (int sig)
Manejador asociado a la señal SIGUSR1.
- void `main` (int argc, char *argv[])

2.2.1 Detailed Description

Ejercicio 4.

Este fichero contiene el código fuente del ejercicio 4 de la entrega.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.2.2 Macro Definition Documentation

2.2.2.1 `#define NUM_ITER 10`

Numero de iteraciones

2.2.3 Function Documentation

2.2.3.1 `bool aredigits (char * string)`

Comprueba si una string es numerica.

Comprueba si cada caracter de la cadena string es un valor numerico.

Parameters

<i>string</i>	Cadena de caracteres.
---------------	-----------------------

Returns

Devuelve true si la cadena es numerica, false si no lo es.

2.2.3.2 void manejador_SIGUSR1 (int sig)

Manejador asociado a la señal SIGUSR1.

Parameters

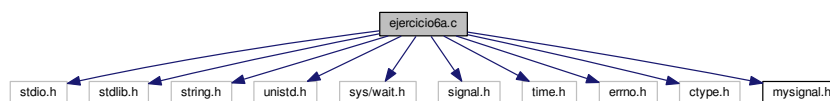
<i>sig</i>	Señal recibida.
------------	-----------------

2.3 ejercicio6a.c File Reference

Ejercicio 6a.

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

Include dependency graph for ejercicio6a.c:



Macros

- #define NUM_PROC 5
- #define SECS 40

Functions

- int main (void)

2.3.1 Detailed Description

Ejercicio 6a.

Este fichero contiene el código fuente del ejercicio 6a de la entrega.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.3.2 Macro Definition Documentation

2.3.2.1 #define NUM_PROC 5

Numero de iteraciones del contador

2.3.2.2 #define SECS 40

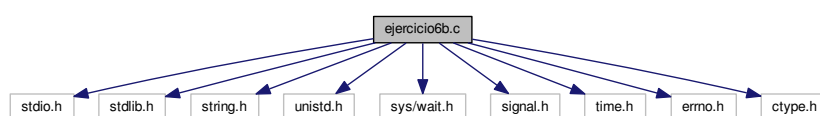
Numero de segundos

2.4 ejercicio6b.c File Reference

Ejercicio 6b.

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

Include dependency graph for ejercicio6b.c:



Macros

- `#define NUM_PROC 5`
- `#define SECS 40`

Functions

- void `manejador_SIGTERM` (int sig)
Manejador asociado a la señal SIGTERM.
- int `main` (void)

2.4.1 Detailed Description

Ejercicio 6b.

Este fichero contiene el código fuente del ejercicio 6b de la entrega.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.4.2 Macro Definition Documentation

2.4.2.1 `#define NUM_PROC 5`

Numero de iteraciones del contador

2.4.2.2 `#define SECS 40`

Numero de segundos

2.4.3 Function Documentation

2.4.3.1 void `manejador_SIGTERM` (int *sig*)

Manejador asociado a la señal SIGTERM.

Parameters

<i>sig</i>	Señal recibida.
------------	-----------------

2.5 ejercicio9.c File Reference

Ejercicio 9.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#include <sys/sem.h>
#include <errno.h>
#include <string.h>
#include <signal.h>
#include <stdbool.h>
#include "semaforos.h"
#include "mysignal.h"
```

Include dependency graph for ejercicio9.c:



Macros

- #define NUM_CAJ 3
- #define NUM_OPER 50
- #define KEY 2018
- #define PATH "/bin/bash"
- #define SIGMONEY SIGRTMIN
- #define SIGDONE SIGRTMIN+1
- #define TEXTDIR "text/"
- #define DATDIR "dat/"
- #define LEE 1
- #define ESCRIBE 0
- #define MAXBUFF 256

Functions

- void cajero (int id)
Dado un id, procesa todas las transacciones asociadas a el.
- bool dinero_recogido (char *filename, bool dinero_sacado, int flag)
Lee de o escribe en el fichero de nombre filename un booleano.
- void handle_SIGDONE (int sig, siginfo_t *info, void *vp)
Manejador de la señal SIGDONE (SIGRTMIN+1)

- void `handle_SIGMONEY` (int sig, siginfo_t *info, void *vp)
Manejador de la señal SIGMONEY (SIGRTMIN).
- float `increase_subtotal` (char *filename, float delta)
Suma delta al float existente en el archivo de nombre filename.
- float `randNum` (float inf, float sup)
Genera un número real aleatorio en el rango [inf, sup)
- int `main` (int argc, char const *argv[])

2.5.1 Detailed Description

Ejercicio 9.

Este fichero contiene el código fuente del ejercicio 9 de la entrega.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.5.2 Macro Definition Documentation

2.5.2.1 #define DATDIR "dat/"

Directorio de los ficheros de datos

2.5.2.2 #define ESCRIBE 0

Flag para dinero_recogido

2.5.2.3 #define KEY 2018

Numero para generar una key con ftok

2.5.2.4 #define LEE 1

Flag para dinero_recogido

2.5.2.5 #define MAXBUFF 256

Tamaño maximo de la cadena de caracteres

2.5.2.6 #define NUM_CAJ 3

Numero de procesos hijo/cajas.

2.5.2.7 #define NUM_OPER 50

Numero de operaciones de cada caja

2.5.2.8 #define PATH "/bin/bash"

Path para generar una key con ftok

2.5.2.9 #define SIGDONE SIGRTMIN+1

Señal para indicar la finalizacion de la caja

2.5.2.10 #define SIGMONEY SIGRTMIN

Señal para indicar de que se tienen mas de 1000€

2.5.2.11 #define TEXTDIR "text/"

Directorio de los ficheros de transacciones

2.5.3 Function Documentation

2.5.3.1 void cajero (int *id*)

Dado un id, procesa todas las transacciones asociadas a el.

Abre su fichero de transacciones correspondiente y las va procesando en otro fichero, avisando al padre cuando supere la cantidad estipulada.

Corresponde a la rutina del proceso hijo.

Parameters

<i>id</i>	Identificador del fichero de transacciones.
-----------	---

2.5.3.2 bool dinero_recogido (char * filename, bool dinero_sacado, int flag)

Lee de o escribe en el fichero de nombre filename un booleano.

Abre el archivo de nombre filename, dependiendo de flag leera un booleano o escribira dinero_sacado. Cierra el fichero.

Parameters

<i>filename</i>	String con el nombre del archivo.
<i>dinero_sacado</i>	Valor bool a escribir. Este parametro se ignora si flag es LEE.
<i>flag</i>	Bandera de valor LEE o ESCRIBE que determinara el comportamiento de la funcion.

Returns

Devuelve el valor del bool de filename si se especifica LEE. Si se especifica ESCRIBE se devuelve false en caso de error, true en otro caso.

2.5.3.3 void handle_SIGDONE (int sig, siginfo_t * info, void * vp)

Manejador de la señal SIGDONE (SIGRTMIN+1)

Retira todo el dinero que quede en la caja del proceso hijo recibido con la señal e incrementa el contador de procesos terminados.

Parameters

<i>sig</i>	Señal recibida.
<i>info</i>	Puntero a la estructura siginfo_t con informacion. Especificada en el manual de sigaction(3).
<i>vp</i>	Puntero a la estructura ucontext_t. Mas informacion en el manual de getcontext(1).

2.5.3.4 void handle_SIGMONEY (int sig, siginfo_t * info, void * vp)

Manejador de la señal SIGMONEY (SIGRTMIN).

Reserva el archivo de donde retirara el dinero pertinente, bloqueandose cuando ya este reservado.

Parameters

<i>sig</i>	Señal recibida.
<i>info</i>	Puntero a la estructura siginfo_t con informacion. Especificada en el manual de sigaction(3).
<i>vp</i>	Puntero a la estructura ucontext_t. Mas informacion en el manual de getcontext(1).

2.5.3.5 float increase_subtotal (char * filename, float delta)

Suma delta al float existente en el archivo de nombre filename.

Se abre el archivo de nombre filename, y se escribe en el el resultado de sumar delta a el float leído del archivo.

Parameters

<i>filename</i>	String con el nombre del archivo.
<i>delta</i>	Valor en floating point a sumar.

Returns

Devuelve ERROR si falla o el float actualizado en exito.

2.5.3.6 float randNum (float *inf*, float *sup*)

Genera un número real aleatorio en el rango [inf, sup)

Genera un número real aleatorio en el rango [inf, sup)

Parameters

<i>inf</i>	Límite inferior del intervalo
<i>sup</i>	Límite superior del intervalo

Returns

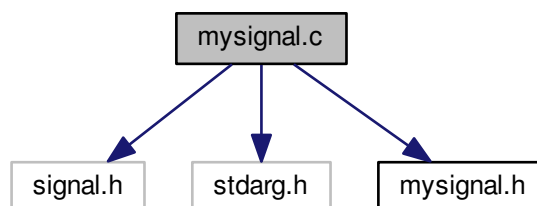
Devuelve el número aleatorio generado

2.6 mysignal.c File Reference

Funciones personales acerca de señales. Implementacion de [mysignal.h](#).

```
#include <signal.h>
#include <stdarg.h>
#include "mysignal.h"
```

Include dependency graph for mysignal.c:



Functions

- void [sigaddset_var](#) (sigset_t *sig, int sig,...)
Añade una lista de señales a un set.

2.6.1 Detailed Description

Funciones personales acerca de señales. Implementacion de [mysignal.h](#).

Este fichero contiene el código fuente de las funciones en la interfaz [mysignal.h](#).

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.6.2 Function Documentation

2.6.2.1 void sigaddset_var (sigset_t * sig, int sig, ...)

Añade una lista de señales a un set.

Añade una lista de señales acabada en -1 al set dado por sig. Ejemplo de uso : sigaddset_var(sig, SIGUSR1, SIGUSR2, SIGINT, -1);

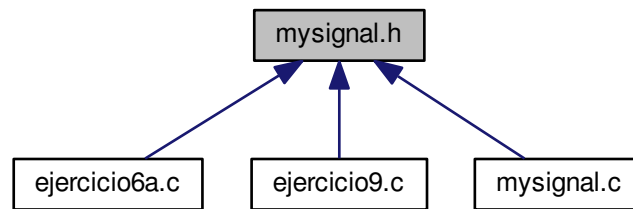
Parameters

<i>sig</i>	Puntero de tipo sigset_t. Set de señales.
<i>sig</i>	Indices de las distintas señales. De extension variable. Debe terminar en -1.

2.7 mysignal.h File Reference

Interfaz de funciones personales acerca de señales.

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



Functions

- void [sigaddset_var](#) (sigset_t *sigt, int sig,...)

Añade una lista de señales a un set.

2.7.1 Detailed Description

Interfaz de funciones personales acerca de señales.

Este fichero contiene las funciones personales acerca de señales.

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.7.2 Function Documentation

2.7.2.1 void sigaddset_var (sigset_t * sigt, int sig, ...)

Añade una lista de señales a un set.

Añade una lista de señales acabada en -1 al set dado por sigt. Ejemplo de uso : sigaddset_var(sigt, SIGUSR1, SIGUSR2, SIGINT, -1);

Parameters

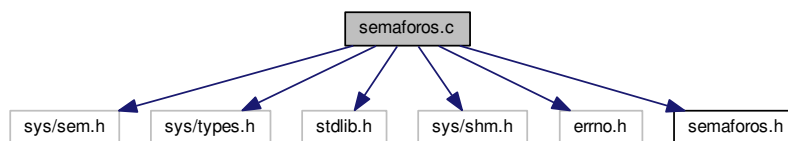
<i>sig</i>	Puntero de tipo sigset_t. Set de señales.
<i>sig</i>	Indices de las distintas señales. De extension variable. Debe terminar en -1.

2.8 semaforos.c File Reference

Funciones personales acerca de semaforos. Implementacion.

```
#include <sys/sem.h>
#include <sys/types.h>
#include <stdlib.h>
#include <sys/shm.h>
#include <errno.h>
#include "semaforos.h"
```

Include dependency graph for semaforos.c:



Functions

- int [inicializar_semaforo](#) (int semid, unsigned short *array)
Inicializa los semaforos indicados.
- int [borrar_semaforo](#) (int semid)
Borra un semáforo.
- int [crear_semaforo](#) (key_t key, int size, int *semid)
Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.
- int [down_semaforo](#) (int semid, int num_sem, int undo)
Baja el semaforo indicado.
- int [down_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Baja todos los semaforos del array indicado por active.
- int [up_semaforo](#) (int semid, int num_sem, int undo)
Sube el semaforo indicado.
- int [up_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Sube todos los semaforos del array indicado por active.

2.8.1 Detailed Description

Funciones personales acerca de semaforos. Implementacion.

Este fichero contiene el código fuente de las funciones en la interfaz [semaforos.h](#).

Author

Rafael Sánchez & Sergio Galán

Version

1.0

Date

06-04-2018

2.8.2 Function Documentation

2.8.2.1 `int borrar_semaforo (int semid)`

Borra un semáforo.

Parameters

<i>semid</i>	Identificador del semaforo.
--------------	-----------------------------

Returns

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

2.8.2.2 `int crear_semaforo (key_t key, int size, int * semid)`

Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.

Parameters

<i>key</i>	Clave precompartida del semaforo.
<i>size</i>	Tamaño del semaforo.
<i>semid</i>	Identificador del semaforo.

Returns

ERROR en caso de error, 0 si ha creado el semaforo, 1 si ya estaba creado.

2.8.2.3 int down_multiple_semaforo (int *semid*, int *size*, int *undo*, int * *active*)

Baja todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

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

2.8.2.4 int down_semaforo (int *semid*, int *num_sem*, int *undo*)

Baja el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

Returns

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

2.8.2.5 int inicializar_semaforo (int *semid*, unsigned short * *array*)

Inicializa 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.

2.8.2.6 int up_multiple_semaforo (int *semid*, int *size*, int *undo*, int * *active*)

Sube todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

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

2.8.2.7 int up_semaforo (int *semid*, int *num_sem*, int *undo*)

Sube el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

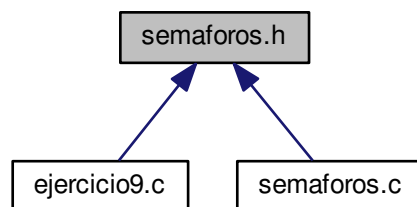
Returns

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

2.9 semaforos.h File Reference

Interfaz de funciones personales acerca de semaforos.

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

**Macros**

- `#define OK EXIT_SUCCESS`
- `#define ERROR -1`

Functions

- int [inicializar_semaforo](#) (int semid, unsigned short *array)
Inicializa los semaforos indicados.
- int [borrar_semaforo](#) (int semid)
Borra un semáforo.
- int [crear_semaforo](#) (key_t key, int size, int *semid)
Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.
- int [down_semaforo](#) (int semid, int num_sem, int undo)
Baja el semaforo indicado.
- int [down_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Baja todos los semaforos del array indicado por active.
- int [up_semaforo](#) (int semid, int num_sem, int undo)
Sube el semaforo indicado.
- int [up_multiple_semaforo](#) (int semid, int size, int undo, int *active)
Sube todos los semaforos del array indicado por active.

2.9.1 Detailed Description

Interfaz de funciones personales acerca de semaforos.

Este fichero contiene las funciones personales acerca de semaforos.

Author

Profesores de SOPER (EDIT: Rafael Sánchez & Sergio Galán).

Version

1.0

Date

06-04-2018

2.9.2 Macro Definition Documentation

2.9.2.1 `#define ERROR -1`

Error

2.9.2.2 `#define OK EXIT_SUCCESS`

Todo fue bien

2.9.3 Function Documentation

2.9.3.1 `int borrar_semaforo (int semid)`

Borra un semáforo.

Parameters

<i>semid</i>	Identificador del semaforo.
--------------	-----------------------------

Returns

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

2.9.3.2 int crear_semaforo (key_t key, int size, int * semid)

Crea un semaforo con la clave y el tamaño especificado. Lo inicializa a 0.

Parameters

<i>key</i>	Clave precompartida del semaforo.
<i>size</i>	Tamaño del semaforo.
<i>semid</i>	Identificador del semaforo.

Returns

ERROR en caso de error, 0 si ha creado el semaforo, 1 si ya estaba creado.

2.9.3.3 int down_multiple_semaforo (int semid, int size, int undo, int * active)

Baja todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

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

2.9.3.4 int down_semaforo (int semid, int num_sem, int undo)

Baja el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

Returns

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

2.9.3.5 int inicializar_semaforo (int *semid*, unsigned short * *array*)

Inicializa 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.

2.9.3.6 int up_multiple_semaforo (int *semid*, int *size*, int *undo*, int * *active*)

Sube todos los semaforos del array indicado por active.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>size</i>	Tamaño de active.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.
<i>active</i>	Semaforos involucrados. Array de enteros.

Returns

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

2.9.3.7 int up_semaforo (int *semid*, int *num_sem*, int *undo*)

Sube el semaforo indicado.

Parameters

<i>semid</i>	Identificador del semaforo.
<i>num_sem</i>	Semaforo dentro del array.
<i>undo</i>	Flag de modo persistente pese a finalización abrupta.

Returns

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

Index

- aredigits
 - [ejercicio4.c](#), [5](#)
- borrar_semaforo
 - [semaforos.c](#), [18](#)
 - [semaforos.h](#), [21](#)
- cajero
 - [ejercicio9.c](#), [11](#)
- crear_semaforo
 - [semaforos.c](#), [18](#)
 - [semaforos.h](#), [22](#)
- DATDIR
 - [ejercicio9.c](#), [10](#)
- dinero_recogido
 - [ejercicio9.c](#), [11](#)
- down_multiple_semaforo
 - [semaforos.c](#), [18](#)
 - [semaforos.h](#), [22](#)
- down_semaforo
 - [semaforos.c](#), [19](#)
 - [semaforos.h](#), [22](#)
- ERROR
 - [semaforos.h](#), [21](#)
- ESCRIBE
 - [ejercicio9.c](#), [10](#)
- [ejercicio2.c](#), [3](#)
 - [MAX_HIJOS](#), [4](#)
- [ejercicio4.c](#), [4](#)
 - [aredigits](#), [5](#)
 - [manejador_SIGUSR1](#), [6](#)
 - [NUM_ITER](#), [5](#)
- [ejercicio6a.c](#), [6](#)
 - [NUM_PROC](#), [7](#)
 - [SECS](#), [7](#)
- [ejercicio6b.c](#), [7](#)
 - [manejador_SIGTERM](#), [8](#)
 - [NUM_PROC](#), [8](#)
 - [SECS](#), [8](#)
- [ejercicio9.c](#), [9](#)
 - [cajero](#), [11](#)
 - [DATDIR](#), [10](#)
 - [dinero_recogido](#), [11](#)
 - [ESCRIBE](#), [10](#)
 - [handle_SIGDONE](#), [12](#)
 - [handle_SIGMONEY](#), [12](#)
 - [increase_subtotal](#), [12](#)
 - [KEY](#), [10](#)
 - [LEE](#), [10](#)
 - [MAXBUFF](#), [10](#)
 - [NUM_CAJ](#), [11](#)
 - [NUM_OPER](#), [11](#)
 - [PATH](#), [11](#)
 - [randNum](#), [14](#)
 - [SIGDONE](#), [11](#)
 - [SIGMONEY](#), [11](#)
 - [TEXTDIR](#), [11](#)
- [handle_SIGDONE](#)
 - [ejercicio9.c](#), [12](#)
- [handle_SIGMONEY](#)
 - [ejercicio9.c](#), [12](#)
- [increase_subtotal](#)
 - [ejercicio9.c](#), [12](#)
- [inicializar_semaforo](#)
 - [semaforos.c](#), [19](#)
 - [semaforos.h](#), [23](#)
- KEY
 - [ejercicio9.c](#), [10](#)
- LEE
 - [ejercicio9.c](#), [10](#)
- MAX_HIJOS
 - [ejercicio2.c](#), [4](#)
- MAXBUFF
 - [ejercicio9.c](#), [10](#)
- [manejador_SIGTERM](#)
 - [ejercicio6b.c](#), [8](#)
- [manejador_SIGUSR1](#)
 - [ejercicio4.c](#), [6](#)
- [mysignal.c](#), [14](#)
 - [sigaddset_var](#), [15](#)
- [mysignal.h](#), [15](#)
 - [sigaddset_var](#), [16](#)
- NUM_CAJ
 - [ejercicio9.c](#), [11](#)
- NUM_ITER
 - [ejercicio4.c](#), [5](#)
- NUM_OPER
 - [ejercicio9.c](#), [11](#)
- NUM_PROC
 - [ejercicio6a.c](#), [7](#)
 - [ejercicio6b.c](#), [8](#)
- OK

- semaforos.h, [21](#)
- PATH
 - ejercicio9.c, [11](#)
- randNum
 - ejercicio9.c, [14](#)
- SECS
 - ejercicio6a.c, [7](#)
 - ejercicio6b.c, [8](#)
- SIGDONE
 - ejercicio9.c, [11](#)
- SIGMONEY
 - ejercicio9.c, [11](#)
- semaforos.c, [17](#)
 - borrar_semaforo, [18](#)
 - crear_semaforo, [18](#)
 - down_multiple_semaforo, [18](#)
 - down_semaforo, [19](#)
 - inicializar_semaforo, [19](#)
 - up_multiple_semaforo, [19](#)
 - up_semaforo, [20](#)
- semaforos.h, [20](#)
 - borrar_semaforo, [21](#)
 - crear_semaforo, [22](#)
 - down_multiple_semaforo, [22](#)
 - down_semaforo, [22](#)
 - ERROR, [21](#)
 - inicializar_semaforo, [23](#)
 - OK, [21](#)
 - up_multiple_semaforo, [23](#)
 - up_semaforo, [23](#)
- sigaddset_var
 - mysignal.c, [15](#)
 - mysignal.h, [16](#)
- TEXTDIR
 - ejercicio9.c, [11](#)
- up_multiple_semaforo
 - semaforos.c, [19](#)
 - semaforos.h, [23](#)
- up_semaforo
 - semaforos.c, [20](#)
 - semaforos.h, [23](#)