Practica 3

Generated by Doxygen 1.8.11

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

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

Chapter 3

Data Structure Documentation

3.1 Data Struct Reference

Estructura para la memoria compartida.

Data Fields

- char **caracter**
- int **contador**

3.1.1 Detailed Description

Estructura para la memoria compartida.

The documentation for this struct was generated from the following file:

- ejercicio3.c

3.2 info Struct Reference

estructura para la memoria compartida

Data Fields

- char **nombre** [80]
- int **id**

3.2.1 Detailed Description

estructura para la memoria compartida

The documentation for this struct was generated from the following files:

- ejercicio2.c
- ejercicio2_solved.c

3.3 Mensaje Struct Reference

Estructura de los mensajes.

Data Fields

- long **mtype**
- int **acabado**
- char **info** [16]

3.3.1 Detailed Description

Estructura de los mensajes.

Compuesta por el tipo, la info y un int que indica si ha acabado

The documentation for this struct was generated from the following file:

- cadena_montaje.c

Chapter 4

File Documentation

4.1 cadena_montaje.c File Reference

Cadena de montaje Ejercicio5.

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for cadena_montaje.c:

4.2 ejercicio2.c File Reference

Ejercicio2.

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

Include dependency graph for ejercicio2.c:

Data Structures

- struct info

  estructura para la memoria compartida

Macros

- #define FILEKEY "/bin/cat"

  file para la clave de ftok
- #define KEY 1300

  clave para el ftok
- #define ALEAT aleat_num(1, 5)

  funcion aleatoria entre 1 y 5 para el sleep de los hijos

Functions

- int aleat_num (int inf, int sup)

  Devuelve un numero aleatorio entre inf y sup En caso de pasar un numero negativo se cambia de signo En caso de que sup sea menor inf se permutan.
- void captura (int sennal)

  Función capturadora de la señal.
- int main (void)

  Main del Ejercicio2.

4.2.1 Detailed Description

Ejercicio2.

Author

Lucia Rivas Molina
Daniel Santo-Tomas Lopez

Date

14/04/2018

4.2.2 Function Documentation

4.2.2.1 int aleat_num (int *inf*, int *sup*)

Devuelve un numero aleatorio entre inf y sup En caso de pasar un numero negativo se cambia de signo En caso de que sup sea menor inf se permutan.

Parameters

inf	minimo numero aleatorio que puede salir
sup	maximo numero aleatorio que puede salir

Returns

el numero aleatorio

4.2.2.2 int main (void)

Main del Ejercicio2.

Returns

EXIT_FAILURE o EXIT_SUCCESS

4.3 ejercicio2_solved.c File Reference

Ejercicio2_solved.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <string.h>
#include <errno.h>
#include <sys/shm.h>
#include <unistd.h>
#include <signal.h>
#include <sys/wait.h>
#include <time.h>
#include <stdlib.h>
#include "semaforos.h"
```

Include dependency graph for ejercicio2_solved.c:

Data Structures

- struct info

  estructura para la memoria compartida

Macros

- #define FILEKEY "/bin/cat"

  file para la clave de ftok
- #define KEY 1300

  clave para el ftok
- #define ALEAT aleat_num(1, 5)

  funcion aleatoria entre 1 y 5 para el sleep de los hijos

Functions

- int aleat_num (int inf, int sup)

  Devuelve un numero aleatorio entre inf y sup En caso de pasar un numero negativo se cambia de signo En caso de que sup sea menor inf se permutan.
- void captura (int sennal)

  Función capturadora de la señal.
- int padre (int semid, int n_hijos, info *inf)

  Funcion que realiza el padre.
- int hijo (int semid, info *inf, int i)

  Funcion que realiza el hijo.
- int main (void)

  Main del Ejercicio2_solved.

4.3.1 Detailed Description

Ejercicio2_solved.

Author

Lucia Rivas Molina
Daniel Santo-Tomas Lopez

Date

14/04/2018

4.3.2 Function Documentation

4.3.2.1 int aleat_num (int *inf,* int *sup*)

Devuelve un numero aleatorio entre inf y sup En caso de pasar un numero negativo se cambia de signo En caso de que sup sea menor inf se permutan.

Parameters

inf	minimo numero aleatorio que puede salir
sup	maximo numero aleatorio que puede salir

Returns

el numero aleatorio

4.3.2.2 void captura (int *sennal*)

Función capturadora de la señal.

Parameters

sennal	: señal mandada En caso de mandar SIGINT el programa se encarga de borrar los semaforos y la memoria compartida

4.3.2.3 int hijo (int *semid,* info * *inf,* int *i*)

Funcion que realiza el hijo.

Parameters

semid	: semid de los semaforos
i	: numero de hijo que es
	inf : estructura de la memoria compartida

Returns

OK o ERROR

4.3.2.4 int main (void)

Main del Ejercicio2_solved.

Returns

EXIT_FAILURE o EXIT_SUCCESS

4.3.2.5 int padre (int *semid*, int *n_hijos*, info * *inf*)

Funcion que realiza el padre.

Parameters

semid	: semid de los semaforos
n_hijos	: numero de hijos que ha creado
	inf : estructura de la memoria compartida

Returns

OK o ERROR

4.4 ejercicio3.c File Reference

Ejercicio3.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <string.h>
#include <errno.h>
#include <sys/shm.h>
#include <unistd.h>
#include <signal.h>
#include <sys/wait.h>
#include <time.h>
#include <stdlib.h>
#include "semaforos.h"
```

Include dependency graph for ejercicio3.c:

Data Structures

- struct Data

  Estructura para la memoria compartida.

Macros

- #define FILEKEY "/bin/cat"

  file para la clave
- #define KEY 1300

  key para los semaforos y la memoria compartida
- #define TAM 36

  tamaño del abecedario y los numeros

Functions

- int productor (int semid, Data *buffer)

  Funcion del productor que produce una letra o numero y la guarda en memoria compartida.
- int consumidor (int semid, Data *buffer)

  Funcion del consumidor que lee la memoria compartida y muestra el caracter por pantalla.
- char caracter (int i)

  Funcion que convierte el caracter correspondiente al entero i pasado como parámetro, las letras o los números dependiendo del valor de la i.
- int main (void)

  Main del ejercicio 3.

4.4.1 Detailed Description

Ejercicio3.

Author

Lucia Rivas Molina
Daniel Santo-Tomas Lopez

Date

14/04/2018

4.4.2 Function Documentation

4.4.2.1 char caracter (int *i*)

Funcion que convierte el caracter correspondiente al entero i pasado como parámetro, las letras o los números dependiendo del valor de la i.

Parameters

i	numero

Returns

-1 en caso de error o el caracter correspondiente a i

4.4.2.2 int consumidor (int *semid,* Data * *buffer*)

Funcion del consumidor que lee la memoria compartida y muestra el caracter por pantalla.

Parameters

semid	: id del semaforo
buffer	: la memoria compartida

Returns

OK o ERROR

4.4.2.3 int main (void)

Main del ejercicio 3.

Returns

EXIT_FAILURE o EXIT_SUCCESS

4.4.2.4 int productor (int *semid,* Data * *buffer*)

Funcion del productor que produce una letra o numero y la guarda en memoria compartida.

Parameters

semid	: id del semaforo
buffer	: la memoria compartida

Returns

OK o ERROR

4.5 ejercicio4.c File Reference

Ejercicio4.

```
#include <sys/mman.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <pthread.h>
#include <time.h>
```

Include dependency graph for ejercicio4.c:

Macros

- #define INF 100

  infimo para el numero aleatorio creado en el fichero
- #define MID 1000

  supremo para el aleatorio creado en el fichero e infimo para el numero de caracteres a escribir
- #define SUP 2000

  supremo para el aleatorio de numero de caracteres a escribir

Functions

- int aleat_num (int inf, int sup)

  Devuelve un numero aleatorio entre inf y sup En caso de pasar un numero negativo se cambia de signo En caso de que sup sea menor inf se permutan.
- void * hilo1 (void *arg)

  Función que realiza el hilo 1 abriendo el fichero arg y llenandolo con una cantidad aleatoria de numeros aleatorios.
- void * hilo2 (void *arg)

  Función que realiza el hilo 2 mapeando el fichero modificado por el hilo 1 y convirtiendo las comas en espacios.
- int main (void)

  Main del ejercicio 4.

4.5.1 Detailed Description

Ejercicio4.

Author

Lucia Rivas Molina
Daniel Santo-Tomas Lopez

Date

14/04/2018

4.5.2 Function Documentation

4.5.2.1 int aleat_num (int *inf*, int *sup*)

Devuelve un numero aleatorio entre inf y sup En caso de pasar un numero negativo se cambia de signo En caso de que sup sea menor inf se permutan.

Parameters

inf	minimo numero aleatorio que puede salir
sup	maximo numero aleatorio que puede salir

Returns

el numero aleatorio

4.5.2.2 void ∗ hilo1 (void ∗ *arg*)

Función que realiza el hilo 1 abriendo el fichero arg y llenandolo con una cantidad aleatoria de numeros aleatorios.

Parameters

arg	: path del fichero a abrir

4.5.2.3 void ∗ hilo2 (void ∗ *arg*)

Función que realiza el hilo 2 mapeando el fichero modificado por el hilo 1 y convirtiendo las comas en espacios.

Parameters

arg	: path del fichero a abrir

4.5.2.4 int main (void)

Main del ejercicio 4.

Returns

EXIT_FAILURE o EXIT_SUCCESS

4.6 semaforos.c File Reference

Biblioteca de semaforos.

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

Include dependency graph for semaforos.c:

Functions

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

  Inicializa los semaforos inficados.
- int Borrar_Semaforo (int semid)

  Elimina los semaforos indicados.
- int Crear_Semaforo (key_t key, int size, int *semid)

  Crea los semaforos indicados.
- int Down_Semaforo (int id, int num_sem, int undo)

  Hace un down de un semaforo.
- int DownMultiple_Semaforo (int id, int size, int undo, int *active)

  Hace un down de un grupo de semaforos.
- int Up_Semaforo (int id, int num_sem, int undo)

  Hace un up de un semaforo.
- int UpMultiple_Semaforo (int id, int size, int undo, int *active)

  Hace un up de un grupo de semaforos.

4.6.1 Detailed Description

Biblioteca de semaforos.

Biblioteca de semaforos para las prácticas de SOPER

Author

Lucia Rivas Molina lucia.rivasmolina@estudiante.uam.es
Daniel Santo-Tomas Lopez daniel.santo-tomas@estudiante.uam.es

Date

20/03/2018

4.6.2 Function Documentation

4.6.2.1 int Borrar_Semaforo (int *semid*)

Elimina los semaforos indicados.

Parameters

semid	: identificador del grupo de semaforos

Returns

OK / ERROR

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

Crea los semaforos indicados.

Parameters

key	: clave precompartida del semaforo
size	: numero de semaforos a crear
semid	: identificador del grupo de semaforos

Returns

OK / ERROR

4.6.2.3 int Down_Semaforo (int *id,* int *num_sem,* int *undo*)

Hace un down de un semaforo.

Parameters

id	: identificador del grupo de semaforos
num_sem	: numero del semaforo del cual hacemos el down
undo	: flags

Returns

OK / ERROR

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

Hace un down de un grupo de semaforos.

Parameters

id	: identificador del grupo de semaforos
size	: numero de semaforos para hacer un down
undo	: flags
active	: semaforos involucrados

Returns

OK / ERROR

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

Inicializa los semaforos inficados.

Parameters

semid	: identificador del grupo de semaforos
array	: valores iniciales del semaforo

Returns

OK / ERROR

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

Hace un up de un semaforo.

Parameters

id	: identificador del grupo de semaforos
num_sem	: numero del semaforo del cual hacemos el down
undo	: flags

Returns

OK / ERROR

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

Hace un up de un grupo de semaforos.

Parameters

id	: identificador del grupo de semaforos
size	: numero de semaforos para hacer un down
undo	: flags
active	: semaforos involucrados

Returns

OK / ERROR

Index