

## Sistemas operativos, Practica 2

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	_Info Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.2	_MyArg Struct Reference . . . . .	5
<b>4</b>	<b>File Documentation</b>	<b>7</b>
4.1	ejer4a.c File Reference . . . . .	7
4.2	ejer4b.c File Reference . . . . .	7
4.2.1	Detailed Description . . . . .	8
4.2.2	Typedef Documentation . . . . .	8
4.2.2.1	Info . . . . .	8
4.2.3	Function Documentation . . . . .	8
4.2.3.1	crearInfo(int dim) . . . . .	8
4.2.3.2	crearMatriz(int dim) . . . . .	8
4.2.3.3	lecturaDeDatos(Info *i1, Info *i2) . . . . .	9
4.2.3.4	liberarInfo(Info *i) . . . . .	9
4.2.3.5	liberarMatriz(int **matriz, int dim) . . . . .	9
4.2.3.6	main() . . . . .	9
4.2.3.7	operacionPrint(void *i1) . . . . .	9

4.3	ejercicio10conSleep.c File Reference . . . . .	10
4.3.1	Detailed Description . . . . .	10
4.3.2	Function Documentation . . . . .	10
4.3.2.1	escribirFichero(FILE *f, char *str) . . . . .	10
4.3.2.2	esperarHijos(int num) . . . . .	11
4.3.2.3	leerFichero(FILE *f, char *str) . . . . .	11
4.3.2.4	main() . . . . .	11
4.4	ejercicio10sincronizado.c File Reference . . . . .	11
4.4.1	Detailed Description . . . . .	12
4.4.2	Function Documentation . . . . .	12
4.4.2.1	escribirFichero(FILE *f, char *str) . . . . .	12
4.4.2.2	esperarHijos(int num) . . . . .	12
4.4.2.3	leerFichero(FILE *f, char *str) . . . . .	12
4.4.2.4	main() . . . . .	12
4.4.2.5	manejador(int sig) . . . . .	13
4.5	ejercicio10sinSleep.c File Reference . . . . .	13
4.5.1	Detailed Description . . . . .	13
4.5.2	Function Documentation . . . . .	13
4.5.2.1	escribirFichero(FILE *f, char *str) . . . . .	13
4.5.2.2	esperarHijos(int num) . . . . .	14
4.5.2.3	leerFichero(FILE *f, char *str) . . . . .	14
4.5.2.4	main() . . . . .	14
4.6	ejercicio3a.c File Reference . . . . .	14
4.6.1	Detailed Description . . . . .	15
4.6.2	Function Documentation . . . . .	15
4.6.2.1	isPrime(int num, int *lista, int tamaño) . . . . .	15
4.6.2.2	main() . . . . .	15
4.6.2.3	nPrimos(void *n) . . . . .	16
4.7	ejercicio3b.c File Reference . . . . .	16
4.7.1	Detailed Description . . . . .	16

4.7.2	Function Documentation	16
4.7.2.1	isPrime(int num, int *lista, int tamanio)	16
4.7.2.2	main()	17
4.7.2.3	nPrimos(int n)	17
4.8	ejercicio6.c File Reference	17
4.8.1	Detailed Description	18
4.8.2	Function Documentation	18
4.8.2.1	main()	18
4.9	ejercicio8.c File Reference	18
4.9.1	Detailed Description	18
4.9.2	Function Documentation	19
4.9.2.1	main(int argc, char **argv)	19
4.9.2.2	mytime()	19
4.9.2.3	sigtermHandler(int sig)	19
4.9.2.4	sigusr1Handler(int sig)	19
<b>Index</b>		<b>21</b>



# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">_Info</a> . . . . .	5
<a href="#">_MyArg</a> . . . . .	5





## Chapter 2

# File Index

### 2.1 File List

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

<a href="#">ejer4a.c</a>	Fichero que contiene ejer4a de la practica 2 de SOPER . . . . .	7
<a href="#">ejer4b.c</a>	Fichero que contiene ejer4b de la practica 2 de SOPER . . . . .	7
<a href="#">ejercicio10conSleep.c</a>	Fichero que contiene el ejer10 de la practica 2 de SOPER . . . . .	10
<a href="#">ejercicio10sincronizado.c</a>	Fichero que contiene una variacion del ejer10 de la practica 2 de SOPER . . . . .	11
<a href="#">ejercicio10sinSleep.c</a>	Fichero que contiene una variacion del ejer10 de la practica 2 de SOPER . . . . .	13
<a href="#">ejercicio3a.c</a>	Fichero que contiene ejer3a de la practica 2 de SOPER para el estudio forks vs threads . . . . .	14
<a href="#">ejercicio3b.c</a>	Fichero que contiene ejer3b de la practica 2 de SOPER para el estudio forks vs threads . . . . .	16
<a href="#">ejercicio6.c</a>	Fichero que contiene ejer6 de la practica 2 de SOPER . . . . .	17
<a href="#">ejercicio8.c</a>	Fichero que contiene ejer8 de la practica 2 de SOPER para el manejo y envio de seniales entre padres e hijos . . . . .	18



## Chapter 3

# Class Documentation

### 3.1 `_Info` Struct Reference

#### Public Attributes

- int **dim**
- int \*\* **matriz**
- int **multiplicador**
- int **numHilo**
- int \* **f1**
- int \* **f2**

#### 3.1.1 Detailed Description

Definimos la estructura `info` que contendra toda la informacion que necesita cada una de las funciones de nuestros hilos.

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

- [ejer4a.c](#)
- [ejer4b.c](#)

### 3.2 `_MyArg` Struct Reference

#### Public Attributes

- int \* **lista**
- int **num**

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

- [ejercicio3a.c](#)



## Chapter 4

# File Documentation

### 4.1 ejer4a.c File Reference

fichero que contiene ej4a de la practica 2 de SOPER

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <string.h>
Include dependency graph for ejer4a.c:
```

### 4.2 ejer4b.c File Reference

fichero que contiene ej4b de la practica 2 de SOPER

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <string.h>
Include dependency graph for ejer4b.c:
```

#### Classes

- struct [\\_Info](#)

#### Typedefs

- typedef struct [\\_Info](#) Info

## Functions

- void `liberarMatriz` (int \*\*matriz, int dim)
- int \*\* `crearMatriz` (int dim)
- int `lecturaDeDatos` (`Info` \*i1, `Info` \*i2)
- `Info` \* `crearInfo` (int dim)
- void `liberarInfo` (`Info` \*i)
- void \* `operacionPrint` (void \*i1)
- int `main` ()

### 4.2.1 Detailed Description

fichero que contiene ej4b de la practica 2 de SOPER

#### Author

Lucia Asencio y Rodrigo de Pool

#### Date

17-3-2017

### 4.2.2 Typedef Documentation

#### 4.2.2.1 typedef struct `_Info` `Info`

Definimos la estructura info que contendra toda la informacion que necesita cada una de las funciones de nuestros hilos.

### 4.2.3 Function Documentation

#### 4.2.3.1 `Info*` `crearInfo` ( int *dim* )

Funcion que se encarga de incilizar la estructura info dada la dimension, tambien inicializa la matriz de dentro

#### Parameters

<i>dim</i>	dimension de la matriz que contiene info
------------	--

#### Returns

Puntero al info creado o null en caso de error

#### 4.2.3.2 `int**` `crearMatriz` ( int *dim* )

Funcion que se encarga de reservar memoria para una matriz

## Parameters

<i>dim</i>	dimesnion de la matriz a la que se le requiere reservar mem
------------	---

## Returns

int \*\* con la matriz creada o NULL en caso de error

4.2.3.3 int lecturaDeDatos ( Info \* *i1*, Info \* *i2* )

Funcion que se encarga de leer los datos del usuario por la linea de comandos en introducirlos en las estructuras Info pasadas como argumento

## Parameters

<i>i1</i>	Estructura Info del primer hilo
<i>i2</i>	Estructura Info del segundo hilo

## Returns

1 si no ha habido error, 0 si ha habido un error

4.2.3.4 void liberarInfo ( Info \* *i* )

Funcion que se encarga de liberar un info

4.2.3.5 void liberarMatriz ( int \*\* *matriz*, int *dim* )

Se encarga de liberar una matriz

## Parameters

<i>matriz</i>	matriz a liberar
<i>dim</i>	la dimension en filas de la matriz

## 4.2.3.6 int main ( )

Funcion principal que ejecutará a través de hilos operaciones matriciales. Nota: Esta funcion puede ser sensible si no se introducen los datos con el formato indicado

4.2.3.7 void\* operacionPrint ( void \* *i1* )

Funcion que ejecutara cada hilo

### 4.3 ejercicio10conSleep.c File Reference

fichero que contiene el ej10 de la practica 2 de SOPER

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

Include dependency graph for ejercicio10conSleep.c:

#### Macros

- #define **NUM** 50
- #define **PATH** "fichero"

#### Functions

- void [escribirFichero](#) (FILE \*f, char \*str)
- void [leerFichero](#) (FILE \*f, char \*str)
- void [esperarHijos](#) (int num)
- int [main](#) ()

#### 4.3.1 Detailed Description

fichero que contiene el ej10 de la practica 2 de SOPER

##### Author

Lucia Asencio y Rodrigo de Pool

##### Date

17-3-2017

#### 4.3.2 Function Documentation

##### 4.3.2.1 void escribirFichero ( FILE \* *f*, char \* *str* )

Se encarga de escribir un string en el fichero

##### Parameters

<i>f</i>	puntero al fichero
<i>str</i>	str string a escribir



4.3.2.2 void esperarHijos ( int *num* )

Funcion que se encarga de esperar a un numero de terminado de hijos

## Parameters

<i>num</i>	numero de hijos
------------	-----------------

4.3.2.3 void leerFichero ( FILE \* *f*, char \* *str* )

Se encarga de leer una linea del fichero y guardarla en str

## Parameters

<i>f</i>	puntero al fichero
<i>str</i>	string donde se guarda el string leído

## 4.3.2.4 int main ( )

Ejercicio 10 propuesto

## 4.4 ejercicio10sincronizado.c File Reference

fichero que contiene una variacion del ej10 de la practica 2 de SOPER

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

Include dependency graph for ejercicio10sincronizado.c:

## Macros

- `#define NUM 50`
- `#define PATH "fichero"`

## Functions

- void [escribirFichero](#) (FILE \**f*, char \**str*)
- void [manejador](#) (int sig)
- void [leerFichero](#) (FILE \**f*, char \**str*)
- void [esperarHijos](#) (int num)
- int [main](#) ()

#### 4.4.1 Detailed Description

fichero que contiene una variacion del ej10 de la practica 2 de SOPER

##### Author

Lucia Asencio y Rodrigo de Pool

##### Date

17-3-2017

#### 4.4.2 Function Documentation

##### 4.4.2.1 void escribirFichero ( FILE \* *f*, char \* *str* )

Se encarga de escribir un string en el fichero

##### Parameters

<i>f</i>	puntero al fichero
<i>str</i>	str string a escribir

##### 4.4.2.2 void esperarHijos ( int *num* )

Funcion que se encarga de esperar a un numero de terminado de hijos

##### Parameters

<i>num</i>	numero de hijos
------------	-----------------

##### 4.4.2.3 void leerFichero ( FILE \* *f*, char \* *str* )

Se encarga de leer una linea del fichero y guardarla en str

##### Parameters

<i>f</i>	puntero al fichero
<i>str</i>	string donde se guarda el string leido

##### 4.4.2.4 int main ( )

Ejercicio 10 propuesto

#### 4.4.2.5 void manejador ( int *sig* )

Esta función imprime que se inicia una 'tanda' de lectura

##### Parameters

<i>sig</i>	[description]
------------	---------------

## 4.5 ejercicio10sinSleep.c File Reference

fichero que contiene una variacion del ej10 de la practica 2 de SOPER

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

Include dependency graph for ejercicio10sinSleep.c:

### Macros

- #define **NUM** 50
- #define **PATH** "fichero"

### Functions

- void [escribirFichero](#) (FILE \*f, char \*str)
- void [leerFichero](#) (FILE \*f, char \*str)
- void [esperarHijos](#) (int num)
- int [main](#) ()

#### 4.5.1 Detailed Description

fichero que contiene una variacion del ej10 de la practica 2 de SOPER

##### Author

Lucia Asencio y Rodrigo de Pool

##### Date

17-3-2017

#### 4.5.2 Function Documentation

##### 4.5.2.1 void escribirFichero ( FILE \* *f*, char \* *str* )

Se encarga de escribir un string en el fichero

**Parameters**

<i>f</i>	puntero al fichero
<i>str</i>	str string a escribir

**4.5.2.2 void esperarHijos ( int *num* )**

Funcion que se encarga de esperar a un numero de terminado de hijos

**Parameters**

<i>num</i>	numero de hijos
------------	-----------------

**4.5.2.3 void leerFichero ( FILE \* *f*, char \* *str* )**

Se encarga de leer una linea del fichero y guardarla en str

**Parameters**

<i>f</i>	puntero al fichero
<i>str</i>	string donde se guarda el string leído

**4.5.2.4 int main ( )**

Ejercicio 10 propuesto

**4.6 ejercicio3a.c File Reference**

fichero que contiene ej3a de la practica 2 de SOPER para el estudio forks vs threads

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <pthread.h>
#include <math.h>
#include <sys/time.h>
#include <assert.h>
Include dependency graph for ejercicio3a.c:
```

**Classes**

- [struct \\_MyArg](#)

## Typedefs

- typedef struct `_MyArg` `MyArg`

## Functions

- int `isPrime` (int num, int \*lista, int tamaño)
- void \* `nPrimos` (void \*n)
- int `main` ()

### 4.6.1 Detailed Description

fichero que contiene ej3a de la practica 2 de SOPER para el estudio forks vs threads

#### Author

Lucia Asencio y Rodrigo de Pool

#### Date

16-3-2017

### 4.6.2 Function Documentation

#### 4.6.2.1 int isPrime ( int num, int \* lista, int tamaño )

Funcion que calcula si un numero es o no primo, cribandolo.

##### Parameters

<i>num</i>	numero a comprobar.
<i>lista</i>	array de enteros con todos los primos anteriores a n
<i>tamaño</i>	numero de elementos de lista

##### Returns

0 si no es primo, 1 en caso contrario

#### 4.6.2.2 int main ( )

Funcion main Genera de manera secuencial 100 hilos, cada uno de los cuales calcula los 10000 numeros primos. Cada hilo se lanzado cuando el anterior completa su ejecucion. Imprime el tiempo total de ejecucion

##### Returns

-1 en caso de error, 0 en caso de exito.

#### 4.6.2.3 void \* nPrimos ( void \* n )

nPrimos: reserva memoria para una lista de tamaño  $n \rightarrow n$ , que asigna a  $n \rightarrow$  lista y devuelve con los  $n$  primeros números primos.

##### Parameters

<i>n</i>	void*, puntero a MyArg, donde $n \rightarrow n > 0$ indica el número de primos a calcular
----------	---

## 4.7 ejercicio3b.c File Reference

fichero que contiene ej3b de la practica 2 de SOPER para el estudio forks vs threads

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include <sys/wait.h>
#include <sys/time.h>
#include <time.h>
#include <assert.h>
Include dependency graph for ejercicio3b.c:
```

### Functions

- int [isPrime](#) (int num, int \*lista, int tamaño)
- int \* [nPrimos](#) (int n)
- int [main](#) ()

#### 4.7.1 Detailed Description

fichero que contiene ej3b de la practica 2 de SOPER para el estudio forks vs threads

##### Author

Lucia Asencio y Rodrigo de Pool

##### Date

16-3-2017

#### 4.7.2 Function Documentation

##### 4.7.2.1 int isPrime ( int num, int \* lista, int tamaño )

Funcion que calcula si un número es o no primo, cribandolo.

**Parameters**

<i>num</i>	numero a comprobar.
<i>lista</i>	array de enteros con todos los primos anteriores a n
<i>tamano</i>	numero de elementos de lista

**Returns**

0 si no es primo, 1 en caso contrario

**4.7.2.2 int main ( )**

Funcion main Genera de manera secuencial 100 hijos, cada uno de los cuales calcula los 10000 numeros primos. Cada hijo se lanzado cuando el anterior completa su ejecucion. Imprime el tiempo total de ejecucion

**Returns**

-1 en caso de error, 0 en caso de exito.

**4.7.2.3 int \* nPrimos ( int n )**

nPrimos: reserva memoria para una lista de tamano n, que devuelve con los n primeros numeros primos.

**Parameters**

<i>n</i>	entero > 0, numero de primos a calcular
----------	---

**Returns**

lista de primos o NULL en caso de error.

## 4.8 ejercicio6.c File Reference

fichero que contiene ej6 de la practica 2 de SOPER

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <string.h>
#include <sys/wait.h>
#include <sys/types.h>
Include dependency graph for ejercicio6.c:
```

**Functions**

- int [main](#) ()

### 4.8.1 Detailed Description

fichero que contiene ej6 de la practica 2 de SOPER

#### Author

Lucia Asencio y Rodrigo de Pool

#### Date

17-3-2017

### 4.8.2 Function Documentation

#### 4.8.2.1 int main ( )

Esta funcion realizara el algoritmo propuesto en el ejercicio 5

#### Returns

1

## 4.9 ejercicio8.c File Reference

fichero que contiene ej8 de la practica 2 de SOPER para el manejo y envio de seniales entre padres e hijos

```
#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>
#include <sys/types.h>
#include <time.h>
Include dependency graph for ejercicio8.c:
```

### Functions

- int [mytime](#) ()
- void [sigtermHandler](#) (int sig)
- void [sigusr1Handler](#) (int sig)
- int [main](#) (int argc, char \*\*argv)

### 4.9.1 Detailed Description

fichero que contiene ej8 de la practica 2 de SOPER para el manejo y envio de seniales entre padres e hijos

#### Author

Lucia Asencio y Rodrigo de Pool

#### Date

16-3-2017



## 4.9.2 Function Documentation

### 4.9.2.1 `int main ( int argc, char ** argv )`

Funcion main: crea N hijos secuencialmente. A partir del ultimo, se pasan entre ellos V veces la senial SIGUSR1, de padres a hijos (siendo el hijo del ultimo hijo el proceso raiz). Despues, repiten el mismo proceso para SIGTERM, antes de hacer exit(). El padre empieza enviando la senial al hijo, y este a su hijo, etc hasta el ultimo hijo, que envia al padre.

#### Parameters

--	--

### 4.9.2.2 `int mytime ( )`

Funcion que devuelve una string con la hora actual

### 4.9.2.3 `void sigtermHandler ( int sig )`

Manejador para la senial SIGTERM

#### Parameters

<i>int</i>	sig senial a manejar
------------	----------------------

### 4.9.2.4 `void sigusr1Handler ( int sig )`

Manejador para SIGUSR1

#### Parameters

<i>sig</i>	senial a manejar
------------	------------------



# Index

- [\\_Info](#), [5](#)
  - [\\_MyArg](#), [5](#)
- [crearInfo](#)
  - [ejer4b.c](#), [8](#)
- [crearMatriz](#)
  - [ejer4b.c](#), [8](#)
- [ejer4a.c](#), [7](#)
- [ejer4b.c](#), [7](#)
  - [crearInfo](#), [8](#)
  - [crearMatriz](#), [8](#)
  - [Info](#), [8](#)
  - [lecturaDeDatos](#), [9](#)
  - [liberarInfo](#), [9](#)
  - [liberarMatriz](#), [9](#)
  - [main](#), [9](#)
  - [operacionPrint](#), [9](#)
- [ejercicio10conSleep.c](#), [10](#)
  - [escribirFichero](#), [10](#)
  - [esperarHijos](#), [11](#)
  - [leerFichero](#), [11](#)
  - [main](#), [11](#)
- [ejercicio10sinSleep.c](#), [13](#)
  - [escribirFichero](#), [13](#)
  - [esperarHijos](#), [14](#)
  - [leerFichero](#), [14](#)
  - [main](#), [14](#)
- [ejercicio10sincronizado.c](#), [11](#)
  - [escribirFichero](#), [12](#)
  - [esperarHijos](#), [12](#)
  - [leerFichero](#), [12](#)
  - [main](#), [12](#)
  - [manejador](#), [12](#)
- [ejercicio3a.c](#), [14](#)
  - [isPrime](#), [15](#)
  - [main](#), [15](#)
  - [nPrimos](#), [15](#)
- [ejercicio3b.c](#), [16](#)
  - [isPrime](#), [16](#)
  - [main](#), [17](#)
  - [nPrimos](#), [17](#)
- [ejercicio6.c](#), [17](#)
  - [main](#), [18](#)
- [ejercicio8.c](#), [18](#)
  - [main](#), [19](#)
  - [mytime](#), [19](#)
  - [sigtermHandler](#), [19](#)
  - [sigusr1Handler](#), [19](#)
- [escribirFichero](#)
  - [ejercicio10conSleep.c](#), [10](#)
  - [ejercicio10sinSleep.c](#), [13](#)
  - [ejercicio10sincronizado.c](#), [12](#)
- [esperarHijos](#)
  - [ejercicio10conSleep.c](#), [11](#)
  - [ejercicio10sinSleep.c](#), [14](#)
  - [ejercicio10sincronizado.c](#), [12](#)
- [Info](#)
  - [ejer4b.c](#), [8](#)
- [isPrime](#)
  - [ejercicio3a.c](#), [15](#)
  - [ejercicio3b.c](#), [16](#)
- [lecturaDeDatos](#)
  - [ejer4b.c](#), [9](#)
- [leerFichero](#)
  - [ejercicio10conSleep.c](#), [11](#)
  - [ejercicio10sinSleep.c](#), [14](#)
  - [ejercicio10sincronizado.c](#), [12](#)
- [liberarInfo](#)
  - [ejer4b.c](#), [9](#)
- [liberarMatriz](#)
  - [ejer4b.c](#), [9](#)
- [main](#)
  - [ejer4b.c](#), [9](#)
  - [ejercicio10conSleep.c](#), [11](#)
  - [ejercicio10sinSleep.c](#), [14](#)
  - [ejercicio10sincronizado.c](#), [12](#)
  - [ejercicio3a.c](#), [15](#)
  - [ejercicio3b.c](#), [17](#)
  - [ejercicio6.c](#), [18](#)
  - [ejercicio8.c](#), [19](#)
- [manejador](#)
  - [ejercicio10sincronizado.c](#), [12](#)
- [mytime](#)
  - [ejercicio8.c](#), [19](#)
- [nPrimos](#)
  - [ejercicio3a.c](#), [15](#)
  - [ejercicio3b.c](#), [17](#)
- [operacionPrint](#)
  - [ejer4b.c](#), [9](#)
- [sigtermHandler](#)
  - [ejercicio8.c](#), [19](#)
- [sigusr1Handler](#)
  - [ejercicio8.c](#), [19](#)