

My Project

Generated by Doxygen 1.8.7

Fri Apr 29 2016 15:35:25

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	_Mensaje Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Data Documentation	5
3.1.2.1	aula	5
3.1.2.2	aviso	5
3.1.2.3	id	5
3.1.2.4	pid	5
3.2	Aula Struct Reference	6
3.2.1	Detailed Description	6
3.2.2	Member Data Documentation	6
3.2.2.1	asientos	6
3.2.2.2	disponible	6
3.2.2.3	total	6
4	File Documentation	7
4.1	alumno.c File Reference	7
4.1.1	Detailed Description	7
4.1.2	Function Documentation	8
4.1.2.1	gestion_alumno	8
4.2	ejercicio1.c File Reference	8
4.2.1	Detailed Description	9
4.2.2	Typedef Documentation	9
4.2.2.1	mensaje	9
4.2.3	Function Documentation	9
4.2.3.1	main	9

4.3	gestion.c File Reference	9
4.3.1	Detailed Description	10
4.3.2	Function Documentation	10
4.3.2.1	main	10
4.4	profesor.c File Reference	11
4.4.1	Detailed Description	11
4.4.2	Function Documentation	12
4.4.2.1	comprobacion_aula	12
4.4.2.2	gestion_profesor	13
4.4.2.3	gestion_profesor_examen	13
4.5	semaforos.h File Reference	13
4.5.1	Detailed Description	14
4.5.2	Typedef Documentation	15
4.5.2.1	mensaje	15
	Index	16

Chapter 1

Class Index

1.1 Class List

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

_Mensaje	Mensaje	5
Aula	Aula	6

Chapter 2

File Index

2.1 File List

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

alumno.c	El alumno del proyecto de la Practica 4 SOPER	7
alumno.h	??
ejercicio1.c	El ejercicio 1 de la Practica 4 SOPER	8
gestion.c	El gestion del proyecto de la Practica 4 SOPER	9
profesor.c	El profesor del proyecto de la Practica 4 SOPER	11
profesor.h	??
semaforos.h	Semaforos.h	13

Chapter 3

Class Documentation

3.1 `_Mensaje` Struct Reference

Mensaje.

```
#include <semaforos.h>
```

Public Attributes

- long `id`
- char `aviso` [4 *1024+1]
- int `pid`
- int `aula`

3.1.1 Detailed Description

Mensaje.

Esta estructura define un mensaje.

3.1.2 Member Data Documentation

3.1.2.1 `int _Mensaje::aula`

`Aula` donde esta asignado el alumno

3.1.2.2 `char _Mensaje::aviso[4 *1024+1]`

Informacion que se quiere transmitir

3.1.2.3 `long _Mensaje::id`

Identificador del mensaje

3.1.2.4 `int _Mensaje::pid`

PID del proceso alumno

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

- [ejercicio1.c](#)
- [semaforos.h](#)

3.2 Aula Struct Reference

[Aula.](#)

```
#include <semaforos.h>
```

Public Attributes

- int [total](#)
- int [disponible](#)
- int * [asientos](#)

3.2.1 Detailed Description

[Aula.](#)

Esta estructura define una aula.

3.2.2 Member Data Documentation

3.2.2.1 int* Aula::asientos

Array de asientos que hay en el aula

3.2.2.2 int Aula::disponible

Alumnos disponibles en el aula

3.2.2.3 int Aula::total

Alumnos totales en el aula

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

- [semaforos.h](#)

Chapter 4

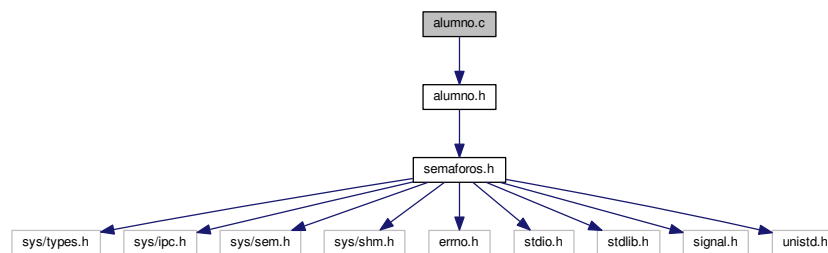
File Documentation

4.1 alumno.c File Reference

El alumno del proyecto de la Practica 4 SOPER.

```
#include "alumno.h"
```

Include dependency graph for alumno.c:



Functions

- void `manejador_SIGTERM` (int sig)
Manejador para SIGTERM Hace que el alumno termine.
- void `realizar_examen` ()
simula realizacion el examen El alumno avisa cuando empeiza, duerme alatoriamente y avisa cuando despierta
- void `gestion_alumno` (int pid, int idCola)
Alumno Gestiona lo que realiza el proceso alumno.

4.1.1 Detailed Description

El alumno del proyecto de la Practica 4 SOPER.

Author

Oscar Garcia de Lara Parreño
Santiago Gomez Aguirre

Version

1.0

Date

29-04-2016

4.1.2 Function Documentation**4.1.2.1 void gestion_alumno (int *pid*, int *id_cola*)**

Alumno Gestiona lo que realiza el proceso alumno.

Parameters

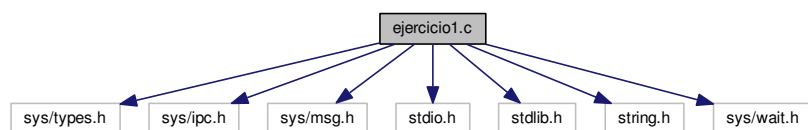
<i>pid</i>	pid del proceso
<i>id_cola</i>	id de la cola de mensajes

4.2 ejercicio1.c File Reference

El ejercicio 1 de la Practica 4 SOPER.

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

Include dependency graph for ejercicio1.c:

**Classes**

- struct [_Mensaje](#)

Mensaje.

Typedefs

- typedef struct [_Mensaje](#) mensaje

Mensaje.

Functions

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

Varios procesos se comunican a traves de colas de mensajes Los procesos se comunican entre si para pasar el contenido de un fichero a otro pero todos los caracteres a mayusculas.

4.2.1 Detailed Description

El ejercicio 1 de la Practica 4 SOPER.

Author

Oscar Garcia de Lara Parreño
Santiago Gomez Aguirre

Version

1.0

Date

29-04-2016

4.2.2 Typedef Documentation

4.2.2.1 typedef struct _Mensaje mensaje

Mensaje.

Esta estructura define un mensaje.

4.2.3 Function Documentation

4.2.3.1 int main (int argc, char const * argv[])

Varios procesos se comunican a traves de colas de mensajes Los procesos se comunican entre si para pasar el contenido de un fichero a otro pero todos los caracteres a mayusculas.

Returns

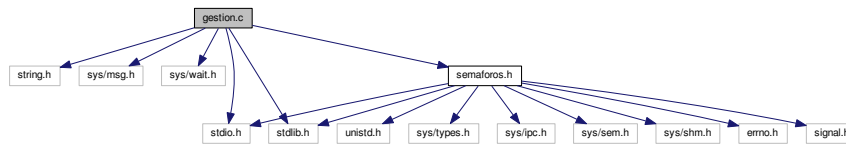
EXIT_FAILURE en caso de error, EXIT_SUCCESS si funciona

4.3 gestion.c File Reference

El gestion del proyecto de la Practica 4 SOPER.

```
#include <string.h>
#include <sys/msg.h>
#include <sys/wait.h>
#include <stdio.h>
#include <stdlib.h>
#include "semaforos.h"
```

Include dependency graph for gestion.c:



Macros

- `#define FILEKEY "/bin/cat"`
- `#define KEY 1300`

Functions

- void `manejador_SIGALRM_30` (int sig)
Manejador para la alarma de 30 segundos Lanza un comando para generar un fichero.
- int `main` (int argc, char *argv[])
Proceso main(Gestion) del simulador Se encargade pedir el numero de asientos y alumnos de la simulacion, y lanzar todos los procesos.

4.3.1 Detailed Description

El gestion del proyecto de la Practica 4 SOPER.

Author

Oscar Garcia de Lara Parreño
Santiago Gomez Aguirre

Version

1.0

Date

29-04-2016

4.3.2 Function Documentation

4.3.2.1 int main (int argc, char * argv[])

Proceso main(Gestion) del simulador Se encargade pedir el numero de asientos y alumnos de la simulacion, y lanzar todos los procesos.

Returns

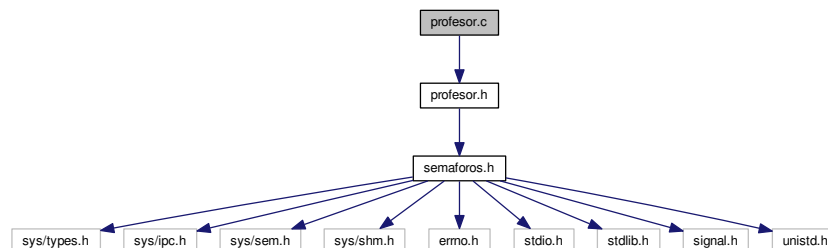
EXIT_FAILURE en caso de error, EXIT_SUCCESS si funciona

4.4 profesor.c File Reference

El profesor del proyecto de la Practica 4 SOPER.

```
#include "profesor.h"
```

Include dependency graph for profesor.c:



Functions

- void `manejador_SIGALRM_5` (int sig)
Manejador para la alarma de 5 minutos Obliga a terminar el examen a los alumnos que lo estan realizando.
- void `gestion_profesor` (int pid, `Aula` *aula, int num_aula, int idCola, int idMutex)
Profesor Gestiona lo que realiza el proceso profesor.
- void `gestion_profesor_examen` (int pid, `Aula` *aula, int num_aula, int idCola, int idMutex)
Profesor_examen Gestiona lo que realiza el proceso profesor_examen.
- int `comprobacion_aula` (`Aula` *aula)
Comprobacion_aula Comprueba que el aula este disponible.

Variables

- `Aula` * `aulag`
- int `num_aulag`
- int `idmutexg`

4.4.1 Detailed Description

El profesor del proyecto de la Practica 4 SOPER.

Author

Oscar Garcia de Lara Parreño
Santiago Gomez Aguirre

Version

1.0

Date

29-04-2016

4.4.2 Function Documentation

4.4.2.1 `int comprobacion_aula (Aula * aula)`

`Comprobacion_aula` Comprueba que el aula este disponible.

Parameters

<i>*aula</i>	array de aulas
--------------	----------------

4.4.2.2 void `gestion_profesor` (int *pid*, Aula * *aula*, int *num_aula*, int *id_cola*, int *idMutex*)

Profesor Gestiona lo que realiza el proceso profesor.

Parameters

<i>pid</i>	pid del proceso
<i>*aula</i>	array de aulas
<i>num_aula</i>	numero del aula en el que esta asignado (0 o 1)
<i>id_cola</i>	id de la cola de mensajes
<i>idMutex</i>	id del semaforo Mutex

4.4.2.3 void `gestion_profesor_examen` (int *pid*, Aula * *aula*, int *num_aula*, int *id_cola*, int *idMutex*)

Profesor_examen Gestiona lo que realiza el proceso profesor_examen.

Parameters

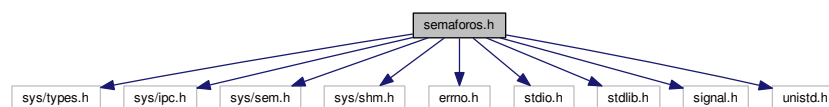
<i>pid</i>	pid del proceso
<i>*aula</i>	array de aulas
<i>num_aula</i>	numero del aula en el que esta asignado (0 o 1)
<i>id_cola</i>	id de la cola de mensajes
<i>idMutex</i>	id del semaforo Mutex

4.5 semaforos.h File Reference

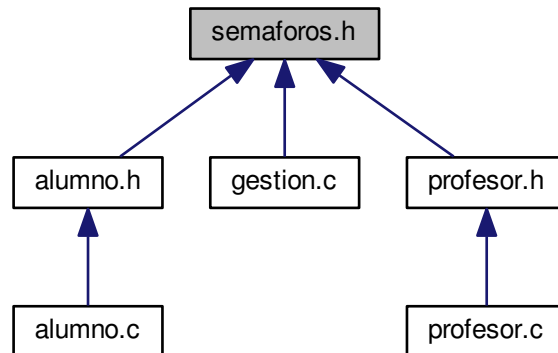
[semaforos.h](#)

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

Include dependency graph for `semaforos.h`:



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



Classes

- struct [_Mensaje](#)
Mensaje.
- struct [Aula](#)
Aula.

Typedefs

- typedef struct [_Mensaje](#) [mensaje](#)
Mensaje.

Enumerations

- enum **STATUS** { **OK** =0, **ERROR** =-1 }

Functions

- int **Inicializar_Semaforo** (int semid, unsigned short *array)
- int **Borrar_Semaforo** (int semid)
- int **Crear_Semaforo** (key_t key, int size, int *semid)
- int **Down_Semaforo** (int id, int num_sem, int undo)
- int **DownMultiple_Semaforo** (int id, int size, int undo, int *active)
- int **Up_Semaforo** (int id, int num_sem, int undo)
- int **UpMultiple_Semaforo** (int id, int size, int undo, int *active)

4.5.1 Detailed Description

[semaforos.h](#)

Author

Oscar Garcia de Lara Parreño
Santiago Gomez Aguirre

Version

1.0

Date

29-03;4-2016

4.5.2 Typedef Documentation**4.5.2.1 typedef struct _Mensaje mensaje**

Mensaje.

Esta estructura define un mensaje.

Index

- asientos
 - Aula, [6](#)
- Aula, [6](#)
 - asientos, [6](#)
 - disponible, [6](#)
 - total, [6](#)
- disponible
 - Aula, [6](#)
- total
 - Aula, [6](#)