

Segunda práctica de Sistemas Operativos

Generado por Doxygen 1.8.14

Índice general

1	Índice de clases	1
1.1	Lista de clases	1
2	Índice de archivos	3
2.1	Lista de archivos	3
3	Documentación de las clases	5
3.1	Referencia de la Unión semun	5
4	Documentación de archivos	7
4.1	Referencia del Archivo aleat_num.h	7
4.1.1	Descripción detallada	7
4.1.2	Documentación de las funciones	7
4.1.2.1	aleat_num()	7
4.2	Referencia del Archivo ejercicio2.c	8
4.2.1	Descripción detallada	8
4.2.2	Documentación de los 'defines'	8
4.2.2.1	NUM_HIJOS	9
4.2.2.2	SEGUNDOS	9
4.2.3	Documentación de las funciones	9
4.2.3.1	main()	9
4.3	Referencia del Archivo ejercicio4.c	9
4.3.1	Descripción detallada	10
4.3.2	Documentación de los 'defines'	10
4.3.2.1	SEGUNDOS	10

4.3.3	Documentación de las funciones	10
4.3.3.1	main()	10
4.3.3.2	manejador()	11
4.4	Referencia del Archivo ejercicio6a.c	11
4.4.1	Descripción detallada	12
4.4.2	Documentación de los 'defines'	12
4.4.2.1	NUM_PROC	12
4.4.2.2	SEGUNDOS	12
4.4.3	Documentación de las funciones	12
4.4.3.1	main()	13
4.5	Referencia del Archivo ejercicio6b.c	13
4.5.1	Descripción detallada	13
4.5.2	Documentación de los 'defines'	14
4.5.2.1	NUM_PROC	14
4.5.2.2	SEGUNDOS	14
4.5.3	Documentación de las funciones	14
4.5.3.1	main()	14
4.5.3.2	manejador()	14
4.6	Referencia del Archivo ejercicio9.c	15
4.6.1	Descripción detallada	16
4.6.2	Documentación de los 'defines'	16
4.6.2.1	FICHERO_SALDO_TOTAL	16
4.6.2.2	KEY	16
4.6.2.3	N_CAJAS	16
4.6.2.4	N_OPERACIONES	16
4.6.2.5	TAM_PATH	17
4.6.2.6	TAMANIO_ARGV_HIJOS	17
4.6.3	Documentación de las funciones	17
4.6.3.1	main()	17
4.6.3.2	manejador_usr1()	17

4.6.3.3	manejador_usr2()	18
4.6.4	Documentación de las variables	18
4.6.4.1	num_procesos_terminados	18
4.6.4.2	oset	18
4.7	Referencia del Archivo ejercicio9hijos.c	18
4.7.1	Descripción detallada	19
4.7.2	Documentación de los 'defines'	19
4.7.2.1	KEY	19
4.7.2.2	MAX_TAM	19
4.7.2.3	N_CAJAS	19
4.7.2.4	SEGUNDOS	20
4.7.3	Documentación de las funciones	20
4.7.3.1	main()	20
4.8	Referencia del Archivo semaforos.h	20
4.8.1	Descripción detallada	21
Índice		23

Capítulo 1

Índice de clases

1.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

semun	5
---------------------------------	---

Capítulo 2

Indice de archivos

2.1. Lista de archivos

Lista de todos los archivos documentados y con descripciones breves:

aleat_num.h	Fichero con la función aleat_num	7
ejercicio2.c	Ejercicio 2 de la Práctica	8
ejercicio4.c	Ejercicio 4 de la Práctica	9
ejercicio6a.c	Ejercicio 6a de la Práctica	11
ejercicio6b.c	Ejercicio 6b de la Práctica	13
ejercicio9.c	Ejercicio 9 de la Práctica	15
ejercicio9hijos.c	Ejercicio 9 de la Práctica (bis)	18
semaforos.h	Fichero proporcionado por los docentes	20

Capítulo 3

Documentación de las clases

3.1. Referencia de la Unión semun

Atributos públicos

- int **val**
- struct semid_ds * **semstat**
- unsigned short * **array**

La documentación para esta unión fue generada a partir del siguiente fichero:

- semaforos.c

Capítulo 4

Documentación de archivos

4.1. Referencia del Archivo aleat_num.h

Fichero con la función aleat_num.

Funciones

- int `aleat_num` (int inf, int sup)
Genera un número aleatorio.

4.1.1. Descripción detallada

Fichero con la función aleat_num.

En este fichero definimos la función aleat_num, ya que es utilizada en más de un fichero, y para así evitar reutilización del código

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.1.2. Documentación de las funciones

4.1.2.1. aleat_num()

```
int aleat_num (  
    int inf,  
    int sup )
```

Genera un número aleatorio.

Esta funcion genera un número aleatorio entre dos especificados

Parámetros

<i>inf</i>	Menor número posible
<i>sup</i>	Mayor número posible

Devuelve

Número generado

4.2. Referencia del Archivo ejercicio2.c

Ejercicio 2 de la Práctica.

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

defines

- #define NUM_HIJOS 4
- #define SEGUNDOS(X) (X) * 1000000

Funciones

- int `main` ()
Función principal del programa.

4.2.1. Descripción detallada

Ejercicio 2 de la Práctica.

En este ejercicio vemos cómo crear hijos, y hacerlos dormir por un número de segundos

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.2.2. Documentación de los 'defines'

4.2.2.1. NUM_HIJOS

```
#define NUM_HIJOS 4
```

Número de hijos a crear

4.2.2.2. SEGUNDOS

```
#define SEGUNDOS(  
    X ) (X) * 1000000
```

Macro para tranformar segundos a microsegundos

4.2.3. Documentación de las funciones

4.2.3.1. main()

```
int main (  
    void )
```

Función principal del programa.

Este programa crea 4 hijos, que imprimirán un mensaje, y dormirán durante 30 segundos. Tras esta espera, volverán a imprimir un mensaje. Mientras tanto, el padre mandará señal de terminar a los hijos tras cinco segundos

Devuelve

0 si todo se ejecuta correctamente, y -1 en cualquier otro caso

4.3. Referencia del Archivo ejercicio4.c

Ejercicio 4 de la Práctica.

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

defines

- #define SEGUNDOS(X) (X) * 1000000

Funciones

- void `manejador` (int senal)
manejador de la señal SIGUSR1
- int `main` (int argc, char **argv)
Función principal del programa.

4.3.1. Descripción detallada

Ejercicio 4 de la Práctica.

En este ejercicio vemos cómo crear una serie de procesos hijos, que tras un tiempo piden al padre ser relevados mediante el uso de señales

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.3.2. Documentación de los 'defines'

4.3.2.1. SEGUNDOS

```
#define SEGUNDOS(  
    X ) (X) * 1000000
```

Macro para tranformar segundos a microsegundos

4.3.3. Documentación de las funciones

4.3.3.1. main()

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

Función principal del programa.

Este programa crea un hijo, que realiza un trabajo (imprimir por pantalla y esperar) 10 veces. Tras esto, manda una señal al padre pidiendo relevo. El padre entonces mata a este proceso y crea otro igual, hasta un máximo de N. Entonces finaliza

Parámetros

<i>argc</i>	numero de parametros de entrada
<i>argv</i>	array de cada parametro. El segundo es el número máximo de procesos a crear

Devuelve

0 si todo se ejecuta correctamente, y -1 en cualquier otro caso

4.3.3.2. manejador()

```
void manejador (  
    int senal )
```

manejador de la señal SIGUSR1

Esta funcion es ejecutada cuando el padre recibe la señal SIGUSR1. Es una función vacía, puesto lo que nos interesa es simplemente esperar a dicha señal

Parámetros

<i>senal</i>	Código de la señal recibida
--------------	-----------------------------

Devuelve

void

4.4. Referencia del Archivo ejercicio6a.c

Ejercicio 6a de la Práctica.

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

defines

- #define NUM_PROC 5
- #define SEGUNDOS 40

Funciones

- `int main (void)`

Función principal del programa.

4.4.1. Descripción detallada

Ejercicio 6a de la Práctica.

En este ejercicio modificamos un código dado, para poner en práctica el uso de alarmas, así como máscaras de bloqueo, y sus diferencias en los bucles

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.4.2. Documentación de los 'defines'

4.4.2.1. NUM_PROC

```
#define NUM_PROC 5
```

Número de procesos a crear

4.4.2.2. SEGUNDOS

```
#define SEGUNDOS 40
```

Número de segundos para la alarma

4.4.3. Documentación de las funciones

4.4.3.1. main()

```
int main (
    void )
```

Función principal del programa.

Este programa crea 5 procesos hijos, que tras ser creado, establecen una alarma. Entonces entran en un bucle, y bloquean la recepción de señales hasta haber imprimido. Por lo tanto, realizan la impresión completa

Devuelve

0 si todo se ejecuta correctamente, y -1 en cualquier otro caso

4.5. Referencia del Archivo ejercicio6b.c

Ejercicio 6b de la Práctica.

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

defines

- #define NUM_PROC 5
- #define SEGUNDOS 40

Funciones

- void `manejador` (int senal)
manejador de la señal SIGTERM
- int `main` (void)
Función principal del programa.

4.5.1. Descripción detallada

Ejercicio 6b de la Práctica.

En este ejercicio modificamos un código dado, para poner en práctica el uso de alarmas, así como máscaras de bloqueo, y sus diferencias en los bucles, en conjunción con el ejercicio 6b

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.5.2. Documentación de los 'defines'

4.5.2.1. NUM_PROC

```
#define NUM_PROC 5
```

Número de procesos a crear

4.5.2.2. SEGUNDOS

```
#define SEGUNDOS 40
```

Número de segundos para la alarma

4.5.3. Documentación de las funciones

4.5.3.1. main()

```
int main (  
    void )
```

Función principal del programa.

Este programa crea 5 procesos hijos, que tras ser creado, establecen una alarma. Entonces entran en un bucle, pero no bloquean las señales, como hacía el [ejercicio6a.c](#) Por lo tanto, se pueden acabar en medio de una impresión

Devuelve

0 si todo se ejecuta correctamente, y -1 en cualquier otro caso

4.5.3.2. manejador()

```
void manejador (  
    int senal )
```

manejador de la señal SIGTERM

Esta funcion es ejecutada cuando el padre recibe la señal SIGTERM. Cuando la recibe, imprime un mensaje y termina su ejecución

Parámetros

<i>senal</i>	Código de la señal recibida
--------------	-----------------------------

Devuelve

void

4.6. Referencia del Archivo ejercicio9.c

Ejercicio 9 de la Práctica.

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

defines

- #define `N_CAJAS` 5
- #define `TAM_PATH` 256
- #define `N_OPERACIONES` 50
- #define `FICHERO_SALDO_TOTAL` "files/saldoTotal.txt"
- #define `TAMANIO_ARGV_HIJOS` 128
- #define `KEY` 15

Funciones

- void `manejador_usr1` (int `senal`)
manejador de la señal SIGRTMIN+1
- void `manejador_usr2` (int `senal`)
manejador de la señal SIGRTMIN
- int `main` ()
Función principal del programa.

Variables

- `sigset_t` `set`
- `sigset_t` `oset`
- int `num_procesos_terminados`

4.6.1. Descripción detallada

Ejercicio 9 de la Práctica.

En este ejercicio creamos una serie de procesos hijos, que realizan cálculos simulando una caja de supermercado, y que mandan señales al proceso padre cuando exceden los 1000 euros o cuando terminan. Se implementa el uso de semáforos

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.6.2. Documentación de los 'defines'

4.6.2.1. FICHERO_SALDO_TOTAL

```
#define FICHERO_SALDO_TOTAL "files/saldoTotal.txt"
```

Path del fichero con el saldo total

4.6.2.2. KEY

```
#define KEY 15
```

Key precompartida por los procesos. Al utilizar exec no podemos usar IPC_PRIVATE

4.6.2.3. N_CAJAS

```
#define N_CAJAS 5
```

Numero de cajas

4.6.2.4. N_OPERACIONES

```
#define N_OPERACIONES 50
```

Numero de operaciones a realizar por los cajeros

4.6.2.5. TAM_PATH

```
#define TAM_PATH 256
```

Maximo tamaño de las path de ficheros

4.6.2.6. TAMANIO_ARGV_HIJOS

```
#define TAMANIO_ARGV_HIJOS 128
```

Tamaño máximo de los argumentos que se pasan a los hijos

4.6.3. Documentación de las funciones

4.6.3.1. main()

```
int main (
    void )
```

Función principal del programa.

Este programa al iniciarse escribe en N_CAJAS ficheros un número de operaciones aleatorias. Después, crea 5 procesos hijos, que ejecutan el programa ejercicio9hijos. Entonces se pone a la espera de las señales SIGRTMIN y SIGRTMIN+1, con las cuales actualiza el saldo total del supermercado

Devuelve

0 si todo se ejecuta correctamente, y -1 en cualquier otro caso

4.6.3.2. manejador_usr1()

```
void manejador_usr1 (
    int senal )
```

manejador de la señal SIGRTMIN+1

Esta funcion es ejecutada cuando el padre recibe la señal SIGRTMIN+1. Esto ocurre cuando un proceso tiene acumulado más de 1000. En ese momento, como el padre no puede saber quién le ha mandado la señal, recorre la lista de ficheros de saldos hasta que encuentra uno con excedentes, y lo actualiza

Parámetros

<i>senal</i>	Código de la señal recibida
--------------	-----------------------------

Devuelve

void

4.6.3.3. manejador_usr2()

```
void manejador_usr2 (
    int senal )
```

manejador de la señal SIGRTMIN

Esta funcion es ejecutada cuando el padre recibe la señal SIGRTMIN. Esto ocurre cuando un proceso ha finalizado su ejecución, tras haber leído todas las operaciones. Como tampoco podemos saber qué proceso la ha lanzado, el padre espera hasta que terminan todos los procesos, y recorre los ficheros actualizando el saldo

Parámetros

<i>senal</i>	Código de la señal recibida
--------------	-----------------------------

Devuelve

void

4.6.4. Documentación de las variables**4.6.4.1. num_procesos_terminados**

```
int num_procesos_terminados
```

Numero de procesos terminados

4.6.4.2. oset

```
sigset_t oset
```

Máscaras de bloqueo y desbloqueo de señales

4.7. Referencia del Archivo ejercicio9hijos.c

Ejercicio 9 de la Práctica (bis)

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <signal.h>
#include <time.h>
#include <unistd.h>
#include "semaforos.h"
#include "aleat_num.h"
```


defines

- #define SEGUNDOS(X) (X) * 1000000
- #define MAX_TAM 256
- #define KEY 15
- #define N_CAJAS 5

Funciones

- int main (int argc, char **argv)
Función principal del programa.

4.7.1. Descripción detallada

Ejercicio 9 de la Práctica (bis)

Este programa es ejecutado por los procesos hijos creados en el ejercicio9. Su ejecución consiste en leer una serie de operaciones de un fichero, y mandar señales al proceso padre, bien cuando exceden un saldo máximo, o cuando han terminado de leer todas las señales

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

4.7.2. Documentación de los 'defines'

4.7.2.1. KEY

```
#define KEY 15
```

Key precompartida de los semáforos

4.7.2.2. MAX_TAM

```
#define MAX_TAM 256
```

Máximo tamaño de las paths

4.7.2.3. N_CAJAS

```
#define N_CAJAS 5
```

Número de cajas

4.7.2.4. SEGUNDOS

```
#define SEGUNDOS(  
    X ) (X) * 1000000
```

Macro para tranformar segundos a microsegundos

4.7.3. Documentación de las funciones

4.7.3.1. main()

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

Función principal del programa.

Este programa lee operaciones de un fichero, y va actualizando otro fichero donde guarda su saldo. Cuando excede los 1000 euros de saldo manda una señal a su padre para que le retire el excedente, y cuando termina, avisa también al padre, para que retire lo que quede de dinero

Devuelve

0 si todo se ejecuta correctamente, y -1 en cualquier otro caso

4.8. Referencia del Archivo semaforos.h

Fichero proporcionado por los docentes.

```
#include <sys/ipc.h>  
#include <sys/sem.h>  
#include <sys/types.h>
```

defines

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

Funciones

- `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)`

4.8.1. Descripción detallada

Fichero proporcionado por los docentes.

En este fichero se definen las cabeceras de las funciones necesarias para utilizar semáforos

Autor

José Manuel Chacón Aguilera y Miguel Arconada Manteca

Fecha

6-4-2018

Índice alfabético

aleat_num
 aleat_num.h, 7
aleat_num.h, 7
 aleat_num, 7

ejercicio2.c, 8
 main, 9
 NUM_HIJOS, 8
 SEGUNDOS, 9

ejercicio4.c, 9
 main, 10
 manejador, 11
 SEGUNDOS, 10

ejercicio6a.c, 11
 main, 12
 NUM_PROC, 12
 SEGUNDOS, 12

ejercicio6b.c, 13
 main, 14
 manejador, 14
 NUM_PROC, 14
 SEGUNDOS, 14

ejercicio9.c, 15
 FICHERO_SALDO_TOTAL, 16
 KEY, 16
 main, 17
 manejador_usr1, 17
 manejador_usr2, 18
 N_CAJAS, 16
 N_OPERACIONES, 16
 num_procesos_terminados, 18
 oset, 18
 TAM_PATH, 16
 TAMANIO_ARGV_HIJOS, 17

ejercicio9hijos.c, 18
 KEY, 19
 MAX_TAM, 19
 main, 20
 N_CAJAS, 19
 SEGUNDOS, 19

FICHERO_SALDO_TOTAL
 ejercicio9.c, 16

KEY
 ejercicio9.c, 16
 ejercicio9hijos.c, 19

MAX_TAM
 ejercicio9hijos.c, 19

main
 ejercicio2.c, 9
 ejercicio4.c, 10
 ejercicio6a.c, 12
 ejercicio6b.c, 14
 ejercicio9.c, 17
 ejercicio9hijos.c, 20

manejador
 ejercicio4.c, 11
 ejercicio6b.c, 14

manejador_usr1
 ejercicio9.c, 17

manejador_usr2
 ejercicio9.c, 18

N_CAJAS
 ejercicio9.c, 16
 ejercicio9hijos.c, 19

N_OPERACIONES
 ejercicio9.c, 16

NUM_HIJOS
 ejercicio2.c, 8

NUM_PROC
 ejercicio6a.c, 12
 ejercicio6b.c, 14

num_procesos_terminados
 ejercicio9.c, 18

oset
 ejercicio9.c, 18

SEGUNDOS
 ejercicio2.c, 9
 ejercicio4.c, 10
 ejercicio6a.c, 12
 ejercicio6b.c, 14
 ejercicio9hijos.c, 19

semaforos.h, 20

semun, 5

TAM_PATH
 ejercicio9.c, 16

TAMANIO_ARGV_HIJOS
 ejercicio9.c, 17