Carrera de caballos

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 Carrera_info Struct Reference

Memoria compartida entre principal, gestor y monitor.

`#include <lib.h>`

Collaboration diagram for Carrera_info:

Data Fields

- int **tiradas** [MAX_CABALLOS]
- int **recorridos** [MAX_CABALLOS]
- int **acabado**
- double **cotizacion** [MAX_CABALLOS]
- int **total_apuestas**
- int **total_caballo** [MAX_CABALLOS]
- int **cont**
- Msg_apuesta **mensaje** [MAX_APOSTADORES]

3.1.1 Detailed Description

Memoria compartida entre principal, gestor y monitor.

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

- lib.h

3.2 Hilo_args Struct Reference

Estructura pasada como argumento a los hilos num_ventana : numero de ventanilla que es id_mem : id de la memoria compartida.

`#include <lib.h>`

Data Fields

- int **num_ventana**
- int **id_cola**
- int **id_mem**

3.2.1 Detailed Description

Estructura pasada como argumento a los hilos num_ventana : numero de ventanilla que es id_mem : id de la memoria compartida.

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

- lib.h

3.3 Msg_apuesta Struct Reference

Mensaje de cada apostador al gestor de apuestas.

```
#include <lib.h>
```

Data Fields

- long **mtype**
- char **nombre** [16]
- double **apuesta**
- int **caballo**
- int **ventanilla**
- double **cotizacion**
- double **pago**

3.3.1 Detailed Description

Mensaje de cada apostador al gestor de apuestas.

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

- lib.h

Chapter 4

File Documentation

4.1 carrera.c File Reference

main de la carrera de caballos

`#include "lib.h"`
Include dependency graph for carrera.c:

4.2 lib.c File Reference

Libreria para las funciones de la carrera entre las que se encuentran las funciones de cada proceso hijo y padre, hilos y auxiliares.

`#include "lib.h"`
Include dependency graph for lib.c:

Functions

- int principal (sigset_t mask, int **pipeIda, int **pipeVuelta, int caballos, int longitud, int semid, Carrera_info *info, pid_t *childpid)

  Funcion del proceso padre del programa, encargado de realizar la comunicacion entre procesos y de asignarle tiradas a los caballos.
- int caballo (int num, int longitud, int *pipeIda, int *pipeVuelta)

  Funcion que realiza el trabajo de los caballos mandando y recibiendo posiciones y recorridos con el padre (principal)
- int monitor (sigset_t mask, int caballos, int longitud, int semid, Carrera_info *info, int dinero)

  Funcion encargada de imprimir por pantalla la informacion.
- int apostador (int dinero, int caballos, int apostadores, int id_cola)

  Funcion encargada de crear apostadores y mandar mensajes a las ventanillas.
- int gestor (sigset_t mask, int id_cola, int caballos, int apostadores, int ventanillas, int id)

  Funcion encargada de inicializar las variables de la memoria compartida relacionadas con el dinero y de crear las ventanillas.
- void * ventanilla (Hilo_args *arg)

  Funcion de los hilos encargada de recibir las apuestas y agregarlas a la memoria compartida.
- void captura (int sennal)

  Funcion capturadora de las señales.
- 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.
- int BubbleSort (int *ganadores, double *pago, int ip, int iu)

  Funcion que ordena la tabla ganadores en funcion de la tabla pago.

4.2.1 Detailed Description

Libreria para las funciones de la carrera entre las que se encuentran las funciones de cada proceso hijo y padre, hilos y auxiliares.

Author

Lucia Rivas Molina
Daniel Santo-Tomas Lopez

Date

9/05/2018

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 apostador (int *dinero*, int *caballos*, int *apostadores*, int *id_cola*)

Funcion encargada de crear apostadores y mandar mensajes a las ventanillas.

Parameters

dinero	: dinero maximo por apostador introducido por teclado
caballos	: numero de caballos
apostadores	: numero de apostadores introducido por teclado
id_cola	: id de la cola de mensajes

Returns

OK o ERROR

4.2.2.3 int BubbleSort (int * *ganadores*, double * *pago*, int *ip*, int *iu*)

Funcion que ordena la tabla ganadores en funcion de la tabla pago.

Se utiliza para los 10 apostadores con mas ganancias

Parameters

ganadores	: tabla de los apostadores en orden de llegada
pago	: tabla del dinero de los apostadores en orden de llegada
ip	: primera iteracion
iu	: ultima iteracion

Returns

numero de iteraciones y ERROR en caso de error

4.2.2.4 int caballo (int *num*, int *longitud*, int * *pipeIda*, int * *pipeVuelta*)

Funcion que realiza el trabajo de los caballos mandando y recibiendo posiciones y recorridos con el padre (principal)

Parameters

num	: numero de caballo que es
pipeIda	: tuberias de escritura con los caballos
pipeVuelta	: tuberias de lectura con los caballos
longitud	: longitud de la carrera

Returns

OK o ERROR

4.2.2.5 void captura (int *sennal*)

Funcion capturadora de las señales.

Parameters

	SIGUSR1 para avisar al padre de que comienza la carrera SIGUSR2 para avisar al gestor y al apostador de que acaben SIGINT para acabar el programa con CTRL+C y borrar memoria

4.2.2.6 int gestor (sigset_t *mask*, int *id_cola*, int *caballos*, int *apostadores*, int *ventanillas*, int *id*)

Funcion encargada de inicializar las variables de la memoria compartida relacionadas con el dinero y de crear las ventanillas.

Parameters

mask	: mascara de señales permitidas
id_cola	: id de la cola de mensajes
caballos	: numero de caballos
apostadores	: numero de apostadores introducido por teclado
ventanillas	: numero de ventanillas introducido por teclado
id	: id de la cola de mensajes

Returns

OK o ERROR

4.2.2.7 int monitor (sigset_t *mask*, int *caballos*, int *longitud*, int *semid*, Carrera_info * *info*, int *dinero*)

Funcion encargada de imprimir por pantalla la informacion.

Parameters

mask	: mascara de señales permitidas
caballos	: numero de caballos
longitud	: longitud de la carrera
semid	: id de los semaforos
info	: estructura con la informacion de la carrera
dinero	: dinero maximo por apostador introducido por teclado

Returns

OK o ERROR

4.2.2.8 int principal (sigset_t *mask*, int ** *pipelda*, int ** *pipeVuelta*, int *caballos*, int *longitud*, int *semid*, Carrera_info * *info*, pid_t * *childpid*)

Funcion del proceso padre del programa, encargado de realizar la comunicacion entre procesos y de asignarle tiradas a los caballos.

Parameters

mask	: mascara de señales permitidas
pipelda	: tuberias de escritura con los caballos
pipeVuelta	: tuberias de lectura con los caballos
caballos	: numero de caballos
longitud	: longitud de la carrera
semid	: id de los semaforos
info	: estructura con la informacion de la carrera
chilpid	: id de los procesos hijos

Returns

OK o ERROR

4.2.2.9 void* ventanilla (Hilo_args * *arg*)

Funcion de los hilos encargada de recibir las apuestas y agregarlas a la memoria compartida.

Parameters

arg	: estructura que contiene el id de la cola y el numero de ventanilla

Returns

pthread_exit

4.3 lib.h File Reference

Libreria para las funciones de la carrera entre las que se encuentran las funciones de cada proceso hijo y padre, hilos y auxiliares.

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

Include dependency graph for lib.h: This graph shows which files directly or indirectly include this file:

Data Structures

- struct Hilo_args

  Estructura pasada como argumento a los hilos num_ventana : numero de ventanilla que es id_mem : id de la memoria compartida.
- struct Msg_apuesta

  Mensaje de cada apostador al gestor de apuestas.
- struct Carrera_info

  Memoria compartida entre principal, gestor y monitor.

Macros

- #define **_GNU_SOURCE**
- #define **LIB_H**
- #define MAX_CABALLOS 10

  maximo numero de caballos
- #define MAX_APOSTADORES 100

  maximo numero de apostadores
- #define FILEKEY "/bin/cat"

  file para la clave
- #define KEY 1300

  key para los semaforos y la memoria compartida

Functions

- int principal (sigset_t mask, int **pipeIda, int **pipeVuelta, int caballos, int longitud, int semid, Carrera_info *info, pid_t *childpid)

  Funcion del proceso padre del programa, encargado de realizar la comunicacion entre procesos y de asignarle tiradas a los caballos.
- int caballo (int num, int longitud, int *pipeIda, int *pipeVuelta)

  Funcion que realiza el trabajo de los caballos mandando y recibiendo posiciones y recorridos con el padre (principal)
- int monitor (sigset_t mask, int caballos, int longitud, int semid, Carrera_info *info, int dinero)

  Funcion encargada de imprimir por pantalla la informacion.
- int apostador (int dinero, int caballos, int apostadores, int id_cola)

  Funcion encargada de crear apostadores y mandar mensajes a las ventanillas.
- int gestor (sigset_t mask, int id_cola, int caballos, int apostadores, int ventanillas, int id)

  Funcion encargada de inicializar las variables de la memoria compartida relacionadas con el dinero y de crear las ventanillas.
- void * ventanilla (Hilo_args *arg)

  Funcion de los hilos encargada de recibir las apuestas y agregarlas a la memoria compartida.
- void captura (int sennal)

  Funcion capturadora de las señales.
- 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.
- int BubbleSort (int *ganadores, double *pago, int ip, int iu)

  Funcion que ordena la tabla ganadores en funcion de la tabla pago.

4.3.1 Detailed Description

Libreria para las funciones de la carrera entre las que se encuentran las funciones de cada proceso hijo y padre, hilos y auxiliares.

Author

Lucia Rivas Molina
Daniel Santo-Tomas Lopez

Date

9/05/2018

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 int apostador (int *dinero*, int *caballos*, int *apostadores*, int *id_cola*)

Funcion encargada de crear apostadores y mandar mensajes a las ventanillas.

Parameters

dinero	: dinero maximo por apostador introducido por teclado
caballos	: numero de caballos
apostadores	: numero de apostadores introducido por teclado
id_cola	: id de la cola de mensajes

Returns

OK o ERROR

4.3.2.3 int BubbleSort (int * *ganadores*, double * *pago*, int *ip*, int *iu*)

Funcion que ordena la tabla ganadores en funcion de la tabla pago.

Se utiliza para los 10 apostadores con mas ganancias

Parameters

ganadores	: tabla de los apostadores en orden de llegada
pago	: tabla del dinero de los apostadores en orden de llegada
ip	: primera iteracion
iu	: ultima iteracion

Returns

numero de iteraciones y ERROR en caso de error

4.3.2.4 int caballo (int *num,* int *longitud,* int * *pipelda,* int * *pipeVuelta*)

Funcion que realiza el trabajo de los caballos mandando y recibiendo posiciones y recorridos con el padre (principal)

Parameters

num	: numero de caballo que es
pipelda	: tuberias de escritura con los caballos
pipeVuelta	: tuberias de lectura con los caballos
longitud	: longitud de la carrera

Returns

OK o ERROR

4.3.2.5 void captura (int *sennal*)

Funcion capturadora de las señales.

Parameters

	SIGUSR1 para avisar al padre de que comienza la carrera SIGUSR2 para avisar al gestor y al apostador de que acaben SIGINT para acabar el programa con CTRL+C y borrar memoria

4.3.2.6 int gestor (sigset_t *mask,* int *id_cola,* int *caballos,* int *apostadores,* int *ventanillas,* int *id*)

Funcion encargada de inicializar las variables de la memoria compartida relacionadas con el dinero y de crear las ventanillas.

Parameters

mask	: mascara de señales permitidas
id_cola	: id de la cola de mensajes
caballos	: numero de caballos
apostadores	: numero de apostadores introducido por teclado
ventanillas	: numero de ventanillas introducido por teclado
id	: id de la cola de mensajes

Returns

OK o ERROR

4.3.2.7 int monitor (sigset_t *mask*, int *caballos*, int *longitud*, int *semid*, Carrera_info * *info*, int *dinero*)

Funcion encargada de imprimir por pantalla la informacion.

Parameters

mask	: mascara de señales permitidas
caballos	: numero de caballos
longitud	: longitud de la carrera
semid	: id de los semaforos
info	: estructura con la informacion de la carrera
dinero	: dinero maximo por apostador introducido por teclado

Returns

OK o ERROR

4.3.2.8 int principal (sigset_t *mask*, int ** *pipelda*, int ** *pipeVuelta*, int *caballos*, int *longitud*, int *semid*, Carrera_info * *info*, pid_t * *childpid*)

Funcion del proceso padre del programa, encargado de realizar la comunicacion entre procesos y de asignarle tiradas a los caballos.

Parameters

mask	: mascara de señales permitidas
pipelda	: tuberias de escritura con los caballos
pipeVuelta	: tuberias de lectura con los caballos
caballos	: numero de caballos
longitud	: longitud de la carrera
semid	: id de los semaforos
info	: estructura con la informacion de la carrera
chilpid	: id de los procesos hijos

Returns

OK o ERROR

4.3.2.9 void* ventanilla (Hilo_args * *arg*)

Funcion de los hilos encargada de recibir las apuestas y agregarlas a la memoria compartida.

Parameters

arg	: estructura que contiene el id de la cola y el numero de ventanilla

Returns

pthread_exit

4.4 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.

4.4.1 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

4.4.2 Function Documentation

4.4.2.1 int Borrar_Semaforo (int *semid*)

Elimina los semaforos indicados.

Parameters

semid	: identificador del grupo de semaforos

Returns

OK / ERROR

4.4.2.2 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

4.4.2.3 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

4.4.2.4 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

4.4.2.5 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

4.4.2.6 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

4.4.2.7 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