



ARQUITECTURA Y SISTEMAS OPERATIVOS II



Práctica Optativa: Procesos

Introducción

En ésta práctica, se trata de completar la implementación de un programa, `7ymedio.c`, para simular la ejecución de N partidas del juego de cartas del “7 y medio” entre el croupier y un jugador. Será un programa compuesto de varios procesos: un proceso Padre (`7ymedio`) que lanza N procesos Hijo (`partida`) de forma consecutiva. El programa `7ymedio` recibirá un único argumento que indicará el número de partidas N que se quieren jugar.

Ficheros Adjuntados: `7ymedio.c`, `partida.c`

Implementación

Programa `7ymedio.c`

Argumentos

Todo programa en C tiene una función `main()` que aparece declarada como:

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

En esta declaración observamos dos parámetros:

- `argc`: entero que contiene el número total de argumentos incluido el nombre del programa
- `argv`: array de punteros a caracteres. Cada elemento del array apunta a la cadena de caracteres correspondiente a cada uno de los argumentos pasados por la línea de comandos. El final del array está señalado por un puntero a `NULL`

Ejercicio 1. Modifica el programa `7ymedio.c` para que la ejecución de la línea de comandos: `_$./7ymedio 3`

muestre correctamente el mensaje que dará el inicio del juego:

```
***Inicio del juego con 3 partidas***
```

Programa `partida.c`

En el fichero `partida.c` se encuentra el código para ejecutar una partida. Se trata de una partida del juego del “7 y medio” entre un croupier y un jugador.

Observa que para no gestionar decimales, en cada carta se guarda su valor multiplicado por 2 (los valores 1 de sota, caballo y rey representan el valor de 0.5).

El código se ha diseñado con un bucle donde, en cada iteración cada jugador: extrae la siguiente carta de la baraja, acumula la puntuación y decide si continúa iterando ó

si acaba (se ha pasado o decide plantarse). Al finalizar se lanza un mensaje a pantalla con el resultado de la partida.

A continuación se muestra cómo sería una posible ejecución de una partida:

```
_$./partida
-Jugador- carta: "7 de Bastos", prev:0.0 actual:7.0
-Croupier- carta: "6 de Copas", prev:0.0 actual:6.0

- Jugador: 7.0 puntos
- Croupier: 6.0 puntos
Jugador, decide plantarse.

-Croupier- carta: "3 de Bastos", prev:6.0 actual:9.0
Croupier se pasa.

- Jugador: 7.0 puntos
- Croupier: 9.0 puntos

Enhorabuena Jugador: has ganado la partida !!!
```

Ejercicio 2. Analiza el código de `partida.c` y estudia su funcionamiento. Prueba a ejecutarlo unas cuantas veces y explica cuál es la utilidad de todas las funciones `srand()` y `rand()` que aparecen en su código.

Ejecución secuencial

Usando un bucle, se pueden ejecutar varias partidas del juego del “7 y medio”. El bucle usará la variable `n` que proviene del primer argumento pasado a `7ymedio` en su línea de comandos (ver el [Ejercicio 1](#)) y creará un proceso Hijo para, de forma secuencial, ejecutar cada una de las partidas (`fork`, `exec1p` ó `execvp`). Al finalizar todas y cada una de las partidas, el programa mostrará por pantalla un mensaje indicando cuántas de ellas han sido ganadas por el croupier.

Ejercicio 3. Realiza todas las modificaciones que consideres necesarias para que el programa `7ymedio.c` lance a ejecutar tantas partidas como el usuario quiera jugar. Al acabarlas debe mostrar por pantalla el número de ellas en las que ha ganado el croupier.

Entrega y Memoria

La entrega de la práctica consiste en el código fuente de cada uno de los ejercicios y una memoria. La memoria debe incluir la respuesta a todos los ejercicios, comentando qué problemas se han encontrado y qué soluciones se han adoptado. La memoria se entregará en un fichero en formato PDF con una extensión máxima de 5 páginas.