



TAREA 2 - Sistemas Operativos

Profesor: Viktor Tapia
Ayudantes: Fabian Saso - Juan Carlos Bustamante
11-10-2016

1. Ejercicio

Crear un programa en C que permita a 2 jugadores jugar una partida de ajedrez. Para esto el programa debe crear un proceso hijo (fork) que se encargara de las jugadas del segundo jugador mientras que el padre maneja las jugadas del primero. Para comunicarse las jugadas los procesos se enviaran mensajes a través de pipes y el tablero debe ser generado utilizando memoria compartida.

Los mensajes de las jugadas deben tener la estructura C8 a D4 donde las letras son las columnas del tablero y los dígitos las filas (inicialmente las piezas se encuentran en las filas 1, 2, 7 y 8). El primer par corresponde al origen y el segundo el destino.

Reglas:

Movimientos: Torre (T): línea recta, cualquier dirección.

Caballo (C): dos espacios en línea recta y luego 1 cambiando de dirección. Puede pasar por encima de las otras piezas.

Alfil (A): en diagonal, cualquier dirección.

Reina (Q): en diagonal y línea recta, cualquier dirección.

Rey (R): en diagonal y línea recta, cualquier dirección. Solo se puede mover un espacio y este no puede estar amenazado.

Peón (P): un espacio hacia delante. Si llegan al final del tablero se pueden convertir en la pieza que diga el jugador (exceptuando al rey). No se puede mover si tiene una pieza al frente.

El jugador puede mover a la torre, el alfil y la reina hasta antes de una de sus propias piezas, hasta una pieza oponente o el final del tablero en la dirección del movimiento.

Ninguna pieza puede ser movida si deja al rey propio en jaque/jaque mate.

Toma de piezas:

El jugador tomara una pieza del oponente si puede colocar en la misma posición que ocupa esta una de sus propias piezas. El peón toma piezas moviéndose en diagonal hacia delante un espacio. Las piezas tomadas salen del tablero.

Jaque/jaque mate:

Un rey se encuentra en jaque si en el siguiente turno del oponente, este puede poner, mediante jugadas legales, una de sus piezas en la posición que usa el rey. Es solo jaque si el jugador puede evitar esa movida moviendo el rey u otra de sus piezas y jaque mate en caso que no tenga opción.

Los alumnos de III, Además deberán implementar el algoritmo Merge sort. Este es un algoritmo de ordenamiento externo estable basado en la técnica divide y vencerás. Su complejidad es $O(n \log n)$. Se implementaran 2 versiones, una clásica y otra mejorada utilizando threads. El objetivo es comparar el tiempo que demoran ambos algoritmos en ordenar listas de distinto largo. Se deben generar 2 ejecutables, cada uno ejecuta 1 de los 2 algoritmos. Los algoritmos leen las listas de números desde un archivo txt (txt de prueba en moodle).



El formato de input y output es el siguiente:

Input:

```
3
4 15 23 84 2 16
12 4 18 23
55 32 -4.02 58 -11
```

Formato input: La primera línea corresponde a la cantidad de listas a ordenar, las siguientes n líneas corresponden a las listas que se ordenaran.

Output:

```
1
Original: 4 15 23 8 42 16
Ordenados: 4 8 15 16 23 42
Tiempo: 10s
2
Original: 12 4 18 23
Ordenados: 4 12 18 23
Tiempo: 4s
3
Original: 55 32 -4.02 58 -11
Ordenados: -11 -4.02 32 55 58
Tiempo: 8s
```

Formato output: La primera línea indica que lista que se esta ordenando, la segunda línea corresponde al orden original de la lista, la tercera línea es la lista ordenada, la cuarta línea es el tiempo que se demoro en ordenar.

2. Ejemplo

```
JuanCarlos@notebook S0/tarea2\$ ./tarea2
Jugando jugador: 1.
  ABCDEFGH
1 TCARQACT
2 PPPPPPPP
3
4
5
6
7 pppppppp
8 tcarqact
```

D2 a D3

```
Jugando jugador: 2.
  ABCDEFGH
1 TCARQACT
2 PPP PPPP
3 P
4
```



```
5
6
7 pppppppp
8 tcarqact
```

3. Condsideraciones

- El programa debe validar si las jugadas son válidas, si los reyes están en jaque o jaque mate, mostrando a los jugadores mensajes por pantalla (salida estándar).
Los mensajes son:

```
Jugada invalida. <causa>.
El jugador # se encuentra en jaque.
El jugador # se encuentra en jaque mate. Fin del juego.
```

- La entrada de las jugadas se leen mediante entrada estándar.
- Las piezas en mayúscula corresponden a las del jugador 1.
- El código tiene que estar bien comentado e indentado.

4. Entrega

- La tarea debe realizarse de manera individual.
- Para la los alumnos de **INF** la fecha de entrega es hasta el **Lunes 31 de Octubre de 2016 a las 23:55 hrs**, mientras que los alumnos de **ILI** la fecha de entrega es hasta el **Viernes 4 de Noviembre de 2016 a las 23:55 hrs**.
- Por cada día de atraso se descuentan 20 puntos (máximo 3 días de atraso).
- La tarea debe subirse a Moodle con el formato **Tarea1_Apellido_Rol.tar.gz**. Dentro del tar debe ir una carpeta con el mismo nombre que contiene el código fuente del programa en c con su makefile. Además del README.txt con las instrucciones de compilación de ser necesario.
- Cualquier intento de copia, **será sancionado con nota 0** para todos los involucrados.