SISTEMAS OPERATIVOS

TRABAJO PRACTICO 2

(TP2)

Curso 2011-2012

INFORMACIÓN IMPORTANTE

Carácter del trabajo práctico 2 (TP2)

La realización del TP2 **no es necesaria** para **aprobar** la asignatura. Por ello, se recuerda al alumno/a la obligación e importancia de hacer el TP2 por sí mismo **sin copiarlo** de otros compañeros, ya que ello repercutirá en perjuicio del propio alumno/a.

Objetivo del TP2

El objetivo de este trabajo es que el alumno implemente el algoritmo de Coffman de detección de interbloqueos estudiado en el Tema 5.

Una de las mejores formas de entender un algoritmo es programándolo, por este motivo el alumno implementará dichos algoritmos haciendo uso del **lenguaje C** aplicado a la creación de programas para el sistema operativo **Linux**.

Medios necesarios para la realización del TP2

Debido a la existencia de una gran variedad de distribuciones de Linux y compiladores de C y para evitar problemas de compatibilidad de plataformas y arquitecturas, el Equipo Docente pone a disposición del alumno una **máquina virtual** con todo lo necesario para la realización de los trabajos así como un **ejemplo de referencia** que define la estructura y el formato del trabajo que el alumno debe entregar.

En principio el alumno puede utilizar para el desarrollo de la práctica cualquier distribución de **Linux** que tenga instalado el compilador **gcc** así como cualquier entorno de desarrollo IDE que considere oportuno. No obstante el Equipo Docente para corregir las prácticas **usará únicamente la máquina virtual en el estado en que es proporcionada** y **no valorará** trabajos que no puedan compilarse o ejecutarse dentro de dicho entorno o que no se ajusten al formato de entrega del ejemplo de referencia.

Por este motivo es muy recomendable que el alumno desarrolle el trabajo en la máquina virtual proporcionada y el ejemplo de referencia y no pierda el tiempo modificando la configuración o reconstruyendo el ejemplo desde cero.

Las instrucciones de descarga e instalación de dicha máquina virtual se encuentran disponibles en la dirección:

`http://www.uned.es/71902048/maquina_virtual.html`

Forma de entregar el TP2

El alumno/a debe enviar un fichero ZIP con la misma estructura que el ejemplo de referencia que contenga un directorio con el código fuente, otro con el programa compilado y un tercero con el informe de la práctica en formato PDF. El nombre del fichero debe ajustarse a la siguiente estructura:

`SO_TP2_Apellido1_Apellido2_Nombre.zip`

Por ejemplo, el alumno Pedro García Escudero debería entregar el siguiente archivo:

`SO_TP2_García_Escudero_Pedro.zip`

Este archivo se debe entregar en el **curso virtual de la asignatura** dentro de la sección **TAREAS**.

Fecha de entrega del TP2

El plazo para entregar el TP2 termina a las **14:00 horas del 23 de enero de 2012**. Esta fecha es **improrrogable**. Los trabajos entregados fuera de plazo no se evaluarán.

Evaluación del TP2

El trabajo consta de dos partes, una *parte básica* en la que se debe implementar el algoritmo exigido y una *parte avanzada* en la cual se llevará a cabo una presentación gráfica de los resultados obtenidos.

- Si el trabajo no se entrega, no compila, no se ejecuta correctamente o no está correctamente documentado se calificará con **0 puntos**.
- Si se implementa la parte básica del trabajo y éste compila, se ejecuta correctamente y está bien documentado se calificará con hasta **5 puntos**.
- Si se implementa y documenta correctamente toda la funcionalidad del trabajo se podrán obtener hasta **10 puntos**.

La nota del TP2 supone un **5 %** de la nota final. Luego la realización completa y perfecta del TP2 supone 0.5 puntos en la nota final, si se tiene **aprobada** la prueba presencial.

¡¡Aviso importante!! El equipo docente se reserva el derecho de ponerse en contacto con el alumno/a y realizarle diferentes cuestiones relativas al TP2 para verificar que efectivamente es el autor del mismo y no lo ha copiado. Si dicha verificación no fuese satisfactoria se bajará a modo de penalización la nota obtenida en el examen.

TP2 – DETECCIÓN DE INTERBLOQUEOS

Aviso Importante

1) Lea el **enunciado completo** del TP2.
2) Antes de ponerse a trabajar y para evitar descuidos es muy recomendable que el alumno **cambie los datos** del ejemplo de referencia (número del trabajo, nombre, apellidos, DNI, centro asociado en el que se ha matriculado y teléfono de contacto) por los suyos propios en todas las partes donde aparezcan (nombre de los archivos y carpetas así como portada del informe de prácticas).

Parte A (elemental)

La parte mínima a desarrollar en este trabajo es un programa en línea de comandos que lleve a cabo las siguientes acciones

1) Preguntar al usuario si desea cargar los datos de entrada manualmente o desde un archivo de texto.

    a. Si se selecciona cargar desde un archivo deberá pedirse el nombre del fichero de entrada y a continuación leerse el contenido del fichero. Dicho fichero será un fichero de texto que contendrá el vector de recursos existentes $\mathbf{R_E}$, la matriz de recursos asignados **A** y la de recursos necesitados adicionales **M** organizados por columnas y separados por espacios. El comienzo de cada una de las matrices está marcado por los separadores "RE=", "A=" y "M=" respectivamente.

    b. Si se selecciona cargar los datos manualmente deberá ir preguntando uno por uno los valores de dichos vectores y matrices.

2) Deberá calcular el vector de recursos disponibles $\mathbf{R_D}$ así como el estado inicial **X**.

3) Deberá aplicar el algoritmo de *Coffman* tal como se describe en el Tema 5 del libro base de la asignatura y determinar la existencia de interbloqueo. Durante el proceso deberá calcular que procesos se van marcando y cuál es el estado **X** después de marcar dicho proceso.

4) Cuando finalice el algoritmo deberá calcular si se ha producido o no interbloqueo, y en caso afirmativo cuales son los procesos que quedan sin marcar.

5) Deberá guardar todos los resultados anteriores en un archivo de texto `salida.txt`

NOTA: Para simplificar la programación se asumirá un **máximo** de 10 procesos, 10 tipos distintos de recursos y 10 unidades de cada tipo de recurso.

EJEMPLO 1 - Supóngase que el fichero de entrada.txt tiene el siguiente contenido

```
RE= 4 3 2
A= 0 1 1
1 1 0
1 0 1
M= 0 0 1
0 2 0
1 1 0
```

Entonces una posible traza de ejecución del algoritmo sería la siguiente:

```
sistemas@SistemasOperativos:~$  ./Trabajo2

Bienvenido al segundo trabajo de Sistemas Operativos
**************************************
Opciones disponibles
-Cargar los datos desde un archivo (pulsar a)
-Cargar datos manualmente (Pulsar m)
**************************************
Introduzca una opción (a,m):
a
Introduzca el nombre del fichero de entrada:
entrada.txt

**************************************
Muchas gracias, sus datos se han cargado correctamente desde el
archivo entrada.txt.
El resultado ha sido guardado en el fichero salida.txt.
**************************************
```

Tras la ejecución del programa debería haberse creado un fichero de texto llamado `salida.txt` cuyo contenido podría ser el siguiente:

```
RESULTADOS DEL ALGORITMO DE COFFMAN
Matrices iniciales
```

```
Re=( 4 3 2 )
A=
0 1 1
1 1 0
1 0 1
M=
0 0 1
0 2 0
1 1 0
Rd=( 2 1 0 )
El estado inicial es X=( 2 1 0 )
Los procesos sin marcar son: p1 p2 p3

El proceso 3 puede ejecutarse, el nuevo estado es
X=( 3 1 1 )
Los procesos sin marcar son: p1 p2

El proceso 1 puede ejecutarse, el nuevo estado es
X=( 3 2 2 )
Los procesos sin marcar son: p2

El proceso 2 puede ejecutarse, el nuevo estado es
X=( 3 2 2 )
No hay procesos sin marcar--> NO HAY INTERBLOQUEO
```

EJEMPLO 2 - Otra posible traza de ejecución sería la siguiente:

```
sistemas@SistemasOperativos:~$  ./Trabajo2

Bienvenido al segundo trabajo de Sistemas Operativos
**************************************
Opciones disponibles
-Cargar los datos desde un archivo (pulsar a)
-Cargar datos manualmente (Pulsar m)
**************************************
Introduzca una opción (a,m):
m

Introduzca el vector de recursos existentes de cada tipo (valores
separados por espacios) y a continuación pulse intro:
4 3 2
```

```
Existen 3 recursos, introduzca el número de procesos:
4

Introduzca la matriz de recursos asignados (4x3) por filas, separando
los elementos de cada columna con espacios y terminado cada fila con
un intro:
0 1 1
1 1 0
1 0 1
1 1 0

Introduzca la matriz de recursos necesitados adicionales a los ya
asignados (4x3) por filas, separando los elementos de cada columna
con espacios y terminado cada fila con un intro:
0 0 1
0 2 0
1 0 0
0 2 1

**************************************
Muchas gracias, sus datos se han procesado correctamente.
El resultado ha sido guardado en el fichero salida.txt.
**************************************
```

En este caso el contenido del fichero `salida.txt` debería ser el siguiente:

```
RESULTADOS DEL ALGORITMO DE COFFMAN
Matrices iniciales
Re=( 4 3 2 )
A=
0 1 1
1 1 0
1 0 1
1 1 0

M=
0 0 1
0 2 0
1 0 0
0 2 1

Rd=( 1 0 0 )
El estado inicial es X=( 1 0 0 )
Los procesos sin marcar son: p1 p2 p3 p4
```

```
El proceso 3 puede ejecutarse, el nuevo estado es
X=( 2 0 1 )
Los procesos sin marcar son: p1 p2 p4

El proceso 1 puede ejecutarse, el nuevo estado es
X=( 2 1 2 )
Los procesos sin marcar son: p2 p4

Ningún proceso sin marcar puede ejecutarse--> HAY INTERBLOQUEO
Los procesos que se bloquean mutuamente son: p2 p4
```

Parte B (avanzada)

En esta parte el alumno deberá desarrollar una interfaz gráfica de usuario GUI que realice las mismas acciones que su contrapartida en línea de comandos, esto es, que permita introducir los el vector $\mathbf{R_E}$, la matriz de recursos asignados **A** y la de recursos necesitados adicionales **M** utilizando tablas o bien cargarlos desde un archivo de texto.

Además de devolver el resultado de aplicar el algoritmo de Coffman en un fichero de texto la interfaz gráfica deberá mostrar el **grafo de asignación de recursos** marcando en el mismo el ciclo que provoca el interbloqueo (en caso de que exista).

La implementación de esta parte es libre, podrán utilizarse las librerías gráficas GTK+ para la interfaz gráfica así como cualquier herramienta de software libre para la representación gráfica del grafo de asignación de recursos.

Nota 1 - En caso de que se desarrolle dicha interfaz gráfica no será necesario implementar la interfaz en línea de comandos.

Nota 2 - Si se utiliza cualquier herramienta o librería de software libre para la creación de la aplicación deberá explicarse y justificarse adecuadamente el porqué de su uso en la memoria del trabajo.

Ayuda - Las siguientes direcciones de internet pueden resultar útiles para la realización de la funcionalidad avanzada:

- Para la interfaz gráfica: http://www.gtk.org/

- Para dibujar el diagrama de asignación de recursos puede resultar muy conveniente utilizar el lenguaje DOT que se describe someramente aquí:

  http://en.wikipedia.org/wiki/DOT_language

- Para interpretar dicho lenguaje se puede usar la herramienta de software libre Graphviz que se encuentra instalada en la máquina virtual:

  http://en.wikipedia.org/wiki/Graphviz

- Véase la siguiente página para una documentación más completa:

  http://www.graphviz.org/Documentation.php