SISTEMAS OPERATIVOS I

Segundo curso Ingeniería Informática. Curso 2004-2005

Práctica 3: Interbloqueo. Realizar un programa que simule el comportamiento de un sistema que utiliza detección y evitación del interbloqueo. (Realizará una u otra simulación dependiendo de cómo sea invocado)

detección: Se invocará al programa de la siguiente manera:

`a.out -d N0 n1 n2 n3 n4 ...`

"-d" especifica detección, N0 el número de procesos para los que se hará la simulación y n1, n2 ... son las instancias disponibles de los distintos tipo de recursos. Ejemplo:

`a.out -d 3 10 5 6 9`

simularía el comportamiento de un sistema que utiliza detección en el que hay 4 tipos de recursos con 10, 5, 6, 9 instancias respectivamente. La simulación se haría para 3 procesos.

evitación: Se invocará al programa de la siguiente manera:

`a.out fich n1 n2 n3 n4 ...`

fich representa el nombre de un fichero en donde están indicadas las necesidades máximas de cada uno de los procesos y n1, n2 ... son las instancias disponibles de los distintos tipos de recursos. Ejemplo:

`a.out pru 10 5 6 9`

simularía el comportamiento de un sistema que utiliza evitación en el que hay 4 tipos de recursos con 10, 5, 6 y 9 instancias respectivamente. Las necesidades máximas estarían en un fichero de nombre *pru*

El formato del fichero de las necesidades es el siguiente:

- Una línea para cada procesos.
- Cada línea contiene las necesidades máximas de cada proceso separadas por uno o más espacios en blanco o tabs. Evidentemente el número de tipos de recurso tiene que coincidir en todos los procesos y con el que se ha pasado como parámetro.
- Las líneas en blanco son ignoradas.

- El número de líneas válidas determina el número de procesos.

Ejemplo de fichero de necesidades máximas en un sistema donde queremos simular 3 procesos y donde hay cuatro tipos de recursos.

```
4  2  1  0
4  0 3    7
0  4   5  2
```

Al iniciarse la simulación ningún proceso tiene nada asignado. El programa irá presentando las siguientes opciones de la simulación:

- **Solicitud recursos por parte de procesos**. La respuesta a esto será asignar al proceso TODOS los recursos que solicita o dejarlo en espera. Los motivos para dejar a un proceso en espera son que no hay suficientes recursos disponibles o que (en el caso de evitación) el sistema quedará en estado inseguro. Se informará de si la solicitud se satisface o si se deja al proceso en espera y del motivo por el que se hace. Obviamente, un proceso en espera no puede solicitar recursos.
- **Liberación de recursos por parte de procesos**. Cuando un proceso libera recursos se comprobará si con los recursos que libera algún proceso de los que están en espera puede salir de la espera. Obviamente, un proceso en espera no puede liberar recursos.
- **información del sistema**. Indicará cuál es el estado de asignación de recursos en el sistema y qué procesos están en espera junto con las solicitudes pendientes. En el caso de simulación del sistema que utiliza detección indicará si hay interbloqueo (para lo cual tendrá que detectarlo) y qué procesos están involucrados en el interbloqueo. En el caso del sistema que usa evitación indicará si el sistema se halla en estado seguro o inseguro (aunque si la simulación está bien hecha el estado siempre será seguro) y una secuencia segura.

FORMA DE ENTREGA Como en la práctica anterior.

FECHA DE ENTREGA VIERNES 28 ENERO 2005