BUS CIRCULAR

1. Introducción

Se pretende simular el funcionamiento de un autobús circular. El autobús dispondrá de un número de asientos y realizará un recorrido circular por un número determinado de paradas a introducir por el usuario. El autobús parará un tiempo en cada parada y tardará un tiempo en realizar el recorrido desde una parada hasta la siguiente.

Los usuarios del autobús generarán de manera totalmente aleatoria la parada en la que desean subir al autobús y la parada en la que se bajarán. Estas paradas han de ser diferentes. En todo momento debe visualizarse la parada en la que se bajará el usuario.

Cuando el autobús llegue a una parada bajarán los pasajeros que tengan dicha parada como punto de bajada y subirán si hay capacidad los que se encuentran esperando en esa parada.

Puede ocurrir que un usuario espere demasiado tiempo a que haya espacio en el autobús, por lo que si transcurrido un determinado tiempo no ha subido al mismo se marchará andando.

Supondremos que un usuario abandona el sistema cuando se ha bajado en su parada o cuando se ha ido andando. La simulación no terminará hasta que todos los usuarios creados hayan abandonado el sistema.

Los valores que deben introducirse por teclado serán:

- Número de usuarios a crear (máximo 50)
- Capacidad del autobús (no más de 8 personas)
- Número de paradas (entre 2 y 6 paradas)
- Tiempo en realizar trayecto entre paradas
- Tiempo de espera en la cola de la parada antes de irse andando.

La simulación deberá mostrar de forma gráfica la solución del enunciado. Para tal fin tan sólo se suministra el servidor gráfico que permite realizar la representación en pantalla.

2. Servidor gráfico

La salida por pantalla se realizará enviando mensajes a una cola de mensajes que será leída por el servidor gráfico. La cola de mensajes deberá crearse con el identificador devuelto por ftok con los siguientes argumentos:

```
ftok("./fichcola.txt", 18);
```

Los mensajes que deben enviarse a la cola de mensajes tienen la siguiente estructura, que está definida en el fichero comun.h:

```
struct tipo_elemento{
      long tipo; //obligatorio para la cola de mensajes
      int pid;
      int parada;
      int inout;
      int pintaborra;
      int destino;
};
```

donde cada uno de los campos representa lo siguiente:

- `tipo` es el tipo de mensajes. El autobús es tipo 1, los clientes son tipo 2.
- `pid` es el pid del proceso que se representa.
- `parada` será la ventana en la que desea representarse. La parada 0 será la del autobús. Las paradas por las que puede pasar el autobús irán desde la 1 hasta como máximo la 6. Esto implica que el número máximo de paradas será 7.
- `inout` nos indica si el cliente llega a la parada o se baja en esa parada. Los posibles valores están definidos en el fichero comun.h y son los siguientes:
  - `IN` visualiza la información en la ventana de llegada de la parada
  - `OUT` visualiza la informacion en la ventana de bajada de la parada
- `pintaborra` será la operación que desee realizarse en pantalla. Los posibles valores están definidos en el fichero comun.h y son los siguientes:
  - `PINTAR` visualiza la información en pantalla
  - `BORRAR` elimina la información de pantalla
- `destino` será la parada de destino

El servidor gráfico, al ser iniciado comprueba si es posible realizar la representación gráfica con la resolución actual de la pantalla. Si la resolución es correcta, envía la señal 10 al proceso que lo creó (su padre) y comienza a procesar los mensajes que lleguen a la cola. Si la resolución no es correcta enviará la señal 12 al proceso que lo creó y finalizará.

Durante la representación, el servidor leerá los mensajes que llegan a la cola e intentará representarlos. Cada vez que un proceso envía un mensaje para dibujarse en una ventana, el servidor gráfico le envía la señal 10 para indicarle que la operación ha sido exitosa. Por tanto, los procesos que envíen mensajes para ser representados en pantalla, deberán recoger la señal devuelta por el servidor gráfico. Las operaciones de borrado no provocan ninguna señal.

El servidor gráfico finalizará cuando reciba la señal 12.

Para facilitar la representación gráfica, una vez creada la cola de mensajes, se podrá hacer uso de la función visualiza, cuyo prototipo es el siguiente:

```
void visualiza(int cola, int parada, int inout, int pintaborra,
                                                 int destino);
```

Esta función visualizará el pid del proceso que la invoca, enviando un mensaje a la cola que figura como primer parámetro (la que conecta con el servidor gráfico) con la información que se pasa en el resto de parámetros.