

| | |
|---|--|
| <p>Tema 1</p> <p>Puente</p> <p>Vencimiento Martes 23 Febrero 18:00 hs</p> | <p>75-08 Sistemas Operativos 2020 2do Cuatrimestre Trabajo Práctico 2</p> |
| <p>Nota: Toda consulta acerca del TP2 debe estar encabezada con las palabras TP2 en el subject y dirigida a lm-7508.C1@fi.uba.ar</p> <p>Pueden usarse para resolver el problema el constructor de un semáforo y los métodos wait() y post(). Pueden usarse clases auxiliares como las vistas en el curso (<i>barrier</i> y <i>switch</i>) El acceso al valor interno del semáforo solo puede usarse para debug.</p> | |

Grupo 1

93644 ASO, SANTIAGO saso@fi.uba.ar
100560 GARCIA, AILEN MAGALI aimgarcia@fi.uba.ar
100701 DJEORDJIAN, ESTEBAN PEDRO edjeordjian@fi.uba.ar

Programar en C++ los programas necesarios para poder correr el escenario descrito a continuación. **Cada programa deberá tener en su fuente e imprimir al lanzarse el nombre y padrón de los integrantes del grupo.** Durante la ejecución se preverán puntos de control para verificar el avance paso a paso de los mismos mediante lecturas desde la terminal. Deberán usarse los nombres y parámetros descritos en el enunciado (los nombres están en **negrita**). Deberán usarse las primitivas de mensajes System V que se encuentran en la página del curso. Los programas se compilarán con el comando *make all* y los ejecutables se borrarán con *make clean*. (o sea, debe programarse un **Makefile** adecuado)

Un puente permite el paso de vehículos en un solo sentido. En ambas cabeceras la ruta es de doble sentido. El puente puede ser usado simultáneamente por múltiples vehículos que vayan en el mismo sentido.

Deben programarse un **vehiculoEW** genérico que cruza de Este a Oeste y un **vehiculoWE** que lo hace de Oeste a Este. Para probarlo se activarán manualmente un número indeterminado de ellos. Los vehiculos deben permitir la ejecución paso a paso. Un **inicializador** y un **finalizador** para comenzar y terminar la simulación y un **terminador** que se encarguen de las estructuras de IPC. Puede ser necesario un **coordinador**. Entregar todos los archivos en un subdirectorio con el nombre del tema (Puente) y comprimido (tgz o zip).

| | |
|---|---|
| <p>Tema 2</p> <p>CRUCE</p> <p>Vencimiento</p> <p>Martes 23 Febrero</p> <p>18:00 hs</p> | <p>75-08 Sistemas Operativos</p> <p>2020 2do Cuatrimestre</p> <p>Trabajo Práctico 2</p> |
| <p>Nota: Toda consulta acerca del TP2 debe estar encabezada con las palabras TP2 en el subject y dirigida a lm-7508.C1@fi.uba.ar</p> <p>Pueden usarse para resolver el problema el constructor de un semáforo y los métodos wait() y post(). Pueden usarse clases auxiliares como las vistas en el curso (<i>barrier</i> y <i>switch</i>) El acceso al valor interno del semáforo solo puede usarse para debug.</p> | |

Grupo 2

99599 MININO, ALAN NAHUEL

nminino@fi.uba.ar

99846 TORRESETTI, LISANDRO

ltorresetti@fi.uba.ar

100073 CAROL LUGONES, LUIS IGNACIO

icarol@fi.uba.ar

Programar en C++ los programas necesarios para poder correr el escenario descrito a continuación. **Cada programa deberá tener en su fuente e imprimir al lanzarse el nombre y padrón de los integrantes del grupo.** Durante la ejecución se preverán puntos de control para verificar el avance paso a paso de los mismos mediante lecturas desde la terminal. Deberán usarse los nombres y parámetros descritos en el enunciado (los nombres están en **negrita**). Deberán usarse las primitivas de mensajes System V que se encuentran en la página del curso. Los programas se compilarán con el comando *make all* y los ejecutables se borrarán con *make clean*. (o sea, debe programarse un **Makefile** adecuado)

Una soga por sobre un precipicio es usada por los monos para cruzar. La soga puede ser usada simultáneamente por múltiples monos siempre que vayan en la misma dirección.

Deben programarse un **simioID** genérico que cruza de Izquierda a Derecha y un **simioDI** que lo hace de Derecha a Izquierda. Para probarlo se activarán manualmente un número indeterminado de ellos. Los simios deben permitir la ejecución paso a paso. Un **inicializador** y un **finalizador** para comenzar y terminar la simulación y un **terminador** que se encarguen de las estructuras de IPC. Puede ser necesario un **coordinador**. Entregar todos los archivos en un subdirectorio con el nombre del tema (Cruce) y comprimido (tgz o zip).

| | |
|---|--|
| <p>Tema 3</p> <p>Ruta</p> <p>Vencimiento Martes 23 Febrero 18:00 hs</p> | <p>75-08 Sistemas Operativos 2020 2do Cuatrimestre Trabajo Práctico 2</p> |
| <p>Nota: Toda consulta acerca del TP2 debe estar encabezada con las palabras TP2 en el subject y dirigida a lm-7508.C1@fi.uba.ar</p> <p>Pueden usarse para resolver el problema el constructor de un semáforo y los métodos wait() y post(). Pueden usarse clases auxiliares como las vistas en el curso (<i>barrier</i> y <i>switch</i>) El acceso al valor interno del semáforo solo puede usarse para debug.</p> | |

Grupo 3

79979 - GONZALEZ, JUAN MANUEL (juagonzalez@fi.uba.ar)

85881 - SILVESTRI, ANDRES (asilvestri@fi.uba.ar)

91076 - PORRAS CARHUAMACA, SHERLY KATERIN (sporras@fi.uba.ar)

97524 - PIZZINI, PATRICIO (ppizzini@fi.uba.ar)

Programar en C++ los programas necesarios para poder correr el escenario descripto a continuación. **Cada programa deberá tener en su fuente e imprimir al lanzarse el nombre y padrón de los integrantes del grupo.** Durante la ejecución se preverán puntos de control para verificar el avance paso a paso de los mismos mediante lecturas desde la terminal. Deberán usarse los nombres y parámetros descriptos en el enunciado (los nombres están en **negrita**). Deberán usarse las primitivas de mensajes System V que se encuentran en la página del curso. Los programas se compilaran con el comando *make all* y los ejecutables se borrarán con *make clean*. (o sea, debe programarse un **Makefile** adecuado)

Una ruta para subir una cuesta con curvas muy cerradas, solo permite su uso en un sentido (Valle al Monte o Monte al Valle). En ambas cabeceras hay vehículos que quieren usarla. La ruta puede ser usada por mas de un vehículo en forma simultánea, siempre que vayan en el mismo sentido

Deben programarse un **vehiculoVM** genérico que y sube del Valle al Monte y un **vehiculoMV** que desciende del monte al valle. Para probarlo se activarán manualmente un número indeterminado de ellos. Los vehiculos deben permitir la ejecución paso a paso. Un **inicializador** y un **finalizador** para comenzar y terminar la simulación y un **terminador** que se encarguen de las estructuras de IPC. Puede ser necesario un **coordinador**. Entregar todos los archivos en un subdirectorio con el nombre del tema (Ruta) y comprimido (tgz o zip).

| | |
|---|--|
| <p>Tema 4</p> <p>Vía Única</p> <p>Vencimiento Martes 23 Febrero 18:00 hs</p> | <p>75-08 Sistemas Operativos 2020 2do Cuatrimestre Trabajo Práctico 2</p> |
| <p>Nota: Toda consulta acerca del TP2 debe estar encabezada con las palabras TP2 en el subject y dirigida a lm-7508.C1@fi.uba.ar</p> <p>Pueden usarse para resolver el problema el constructor de un semáforo y los métodos wait() y post(). Pueden usarse clases auxiliares como las vistas en el curso (<i>barrier</i> y <i>switch</i>) El acceso al valor interno del semáforo solo puede usarse para debug.</p> | |

Grupo 4

76624 CAMACHO OCAMPO, ISAAC EDGAR icamacho@fi.uba.ar

96240 FELIZ, PABLO HECTOR pfeliz@fi.uba.ar

96457 BAZZANA, MATIAS ESTEBAN mbazzana@fi.uba.ar

99535 LEGUIZAMON, AGUSTIN MANUEL aleguizamon@fi.uba.ar

Programar en C++ los programas necesarios para poder correr el escenario descrito a continuación. **Cada programa deberá tener en su fuente e imprimir al lanzarse el nombre y padrón de los integrantes del grupo.** Durante la ejecución se preverán puntos de control para verificar el avance paso a paso de los mismos mediante lecturas desde la terminal. Deberán usarse los nombres y parámetros descriptos en el enunciado (los nombres están en **negrita**). Deberán usarse las primitivas de mensajes System V que se encuentran en la página del curso. Los programas se compilarn con el comando *make all* y los ejecutables se borrarán con *make clean*. (o sea, debe programarse un **Makefile** adecuado)

Un paso de montaña tiene una única vía férrea que conecta las cabeceras Norte y Sur. Ambas cabeceras continúan con doble vía. Si la vía está habilitada en un sentido, por ella pueden transitar múltiples formaciones en ese sentido.

Deben programarse un **trenNS** genérico que va de la Norte al Sur y un **trenSN** que va de Sur a Norte. Para probarlo se activarán manualmente un número indeterminado de ellos. Los trenes deben permitir la ejecución paso a paso. Un **inicializador** y un **finalizador** para comenzar y terminar la simulación y un **terminador** que se encarguen de las estructuras de IPC. Puede ser necesario un **coordinador**. Entregar todos los archivos en un subdirectorío con el nombre del tema (Via_unica) y comprimido (tgz o zip).