

## I OBJETIVOS

En esta práctica el alumno:

- Utilizará los semáforos como una herramienta para la sincronización de procesos concurrentes.
- Conocerá y usará algunos mecanismos de comunicación entre procesos que provee el sistema operativo tales como los buzones de mensajes.

## II BIBLIOGRAFÍA

- William Stallings, “SISTEMAS OPERATIVOS”, Prentice Hall.
- Silberschatz, Galvin, Gagne, “SISTEMAS OPERATIVOS”, Mc Graw Hill.
- Neil Matthew & Richard Stones, “BEGINNING LINUX PROGRAMMING”, Wrox.

## III RECURSOS

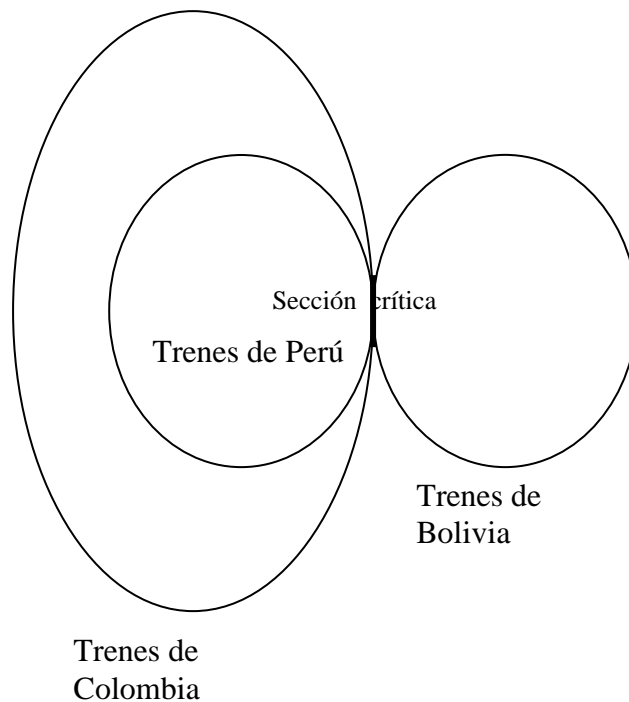
- Una estación de trabajo con Linux con su ambiente gráfico.
- Un editor de texto.
- El compilador de C GNU.

## IV ACTIVIDADES

### 1 *Descripción del problema*

En los países sudamericanos Perú, Bolivia y Colombia existe una red de narcotraficantes que utilizando ferrocarriles transportan cocaína a sus clientes que están en el otro lado de la cordillera de los Andes.

En la cordillera de los Andes existe un tramo muy angosto en el cual las vías de los tres ferrocarriles se unen en una y sólo es posible que pase un ferrocarril a la vez (ver Figura 1). Para esto, los narcos decidieron probar las soluciones que provee el sistema operativo como Semáforos y Colas de Mensajes.



**Figura 1.- Ilustración de la sección crítica en el problema de los trenes.**

## 1.1 La solución con procesos

En el Ejemplo 1 mostramos el problema de los ferrocarriles de los narcos, representados por procesos.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define CICLOS 10

char *pais[3]={"Peru", "Bolvia", "Colombia"};

int *g;

void proceso(int i)
{
    int k;
    int l;

    for(k=0; k<CICLOS; k++)
    {
```

```
// Entrada a la sección crítica
printf("Entra %s",pais[i]);
fflush(stdout);
sleep(rand()%3);
printf("- %s Sale\n",pais[i]);
// Salida de la sección crítica

// Espera aleatoria fuera de la sección crítica
sleep(rand()%3);
}
exit(0);    // Termina el proceso
}

int main()
{
    int pid;
    int status;
    int args[3];
    int i;

    srand(getpid());

    for(i=0;i<3;i++)
    {
        // Crea un nuevo proceso hijo que ejecuta la función proceso()
        pid=fork();
        if(pid==0)
            proceso(i);
    }

    for(i=0;i<3;i++)
        pid = wait(&status);
}
```

**Ejemplo 1. El problema de los trenes sin sincronización.**

## 2 Solución con semáforos

Utilizando los semáforos que provee el sistema operativo Linux (Tipo UNIX) protege las secciones críticas de los procesos que representan cada uno de los trenes.

Las llamadas que provee el sistema operativo para pedir y usar semáforos son de bajo nivel (`semget`, `semop`, `semctl`), por lo que es necesario que definas en un archivo llamado `semaphores.h` funciones que faciliten el desarrollo de aplicaciones con semáforos, tales como:

- `seminit(int idsem, int val)`: para inicializar el semáforo con un valor.
- `semwait(int idsem)`: decrementar el semáforo.

- `sem_signal(int idsem):` incrementar el semáforo.

### **3 Solución con buzones de mensajes**

Utilizando las colas de mensajes o buzones que provee el sistema operativo Linux (Tipo UNIX) protege las secciones críticas de los procesos que representan cada uno de los trenes.

Las llamadas que provee el sistema operativo para pedir y usar colas de mensajes son (`msgget`, `msgsnd`, `msgrcv`)

## V ENTREGA Y EVALUACIÓN



**No incluya líneas de código en sus programas de las cuales desconozca su funcionamiento. El código no conocido será anulado en el funcionamiento de la práctica.**



**Aunque en semestres anteriores se han realizado prácticas similares a esta, hay aspectos que hacen que esta sea diferente. Cualquier evidencia que muestre el intento de entregar una práctica de semestre anterior será calificada como plagio.**

### 1 Entrega y Revisión

Entregar en el apartado correspondiente de Moodle los programas fuentes con su respectivo Makefile en un archivo ZIP a más tardar el Jueves 30 de Octubre a las 11:59 PM. La revisión se realizará durante la semana 12.

### 2 Equipos

Esta práctica se hará en equipos (máximo 3 integrantes), es necesario que en la revisión esté el equipo completo ya que el integrante que no se presente no tendrá calificación en la práctica.

Importante: Al indicarse que el trabajo debe ser desarrollado por equipos, se entiende que no se permite colaboración entre equipos, cualquier evidencia de esto será considerada plagio.

### 3 Evaluación

Puntualidad en las revisiones	El equipo estuvo completo y puntual en todas las sesiones de revisión.	Si hubo dos o más sesiones con el equipo, el equipo estuvo completo y puntual en casi todas las sesiones de revisión	Si solo hubo una sesión de revisión, el equipo no estuvo completo o no fue puntual. Si fueron dos o más sesiones de revisión, en más de una sesión el equipo no estuvo completo o fue puntual
	+10	+5	0
Funcionamiento	El producto cumple con todas las especificaciones indicadas en el documento y no tiene fallas	El producto muestra una falla no esperada o el producto está casi completo, puede funcionar excepto la parte no completada	El producto muestra más de una falla inesperada o no funciona, esto puede ser debido a que no esté completo.
	+75	+37.5	0
Interfaz con el usuario	El producto funciona y pudo ser utilizado sin necesidad de recibir indicaciones por el desarrollador, tiene instrucciones claras para ser utilizado.	El producto funciona, pero hubo necesidad de recibir alguna indicación para su uso por parte del desarrollador del producto	El producto carece de instrucciones claras para ser utilizado y requiere que alguno de los desarrolladores esté presente para su utilización o no puede utilizarse debido a que no está completo
	+10	+5	0
Claridad en el código	El código es claro, usa nombres de variables adecuadas, está debidamente comentado e indentado. Puede ser entendido por cualquier otra persona que no intervino en su desarrollo.	El código carece de claridad, puede ser entendido por cualquier persona ajena a su desarrollo pero con cierta dificultad.	El código carece de comentarios, está mal indentado, usa nombres de variables no adecuadas.
	+5	+2.5	0
Defensa del producto	Todos los integrantes son capaces de explicar cualquier parte del producto presentado	Alguno de los integrantes muestra dudas sobre alguna parte del desarrollo del producto presentado	Más de un integrante, o si el trabajo fue individual, el desarrollador duda sobre cómo está desarrollado producto.
	x 1 (puntos se multiplican por 1)	x 0.5 (puntos se multiplican por 0.5)	x 0 (puntos se multiplican por 0)
Sobresaliente 20 %	Tiene 1 en todos los puntos anteriores. El producto entregado es sobresaliente, muestra tener la calidad para ser expuesto como un producto representativo de la carrera Hay evidencia de que los desarrolladores se documentaron y muestran aprendizajes más allá de lo esperado		No tiene 1 en todos los puntos anteriores, o el producto entregado no es sobresaliente y no muestra tener la calidad para ser expuesto como un producto representativo de la carrera o no hay evidencia de que los desarrolladores se documentaron y muestran aprendizajes más allá de lo esperado
	+20		0