**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**ASIGNATURA**

SISTEMAS OPERATIVOS I

**DOCENTE**

DRA. CARMEN CERON GARNICA

**INTEGRANTES DEL EQUIPO:**

MARIELA SANDOVAL SANTILLÁN (202160368)

KAREN PAOLA PAREDES MARTINEZ (202140741)

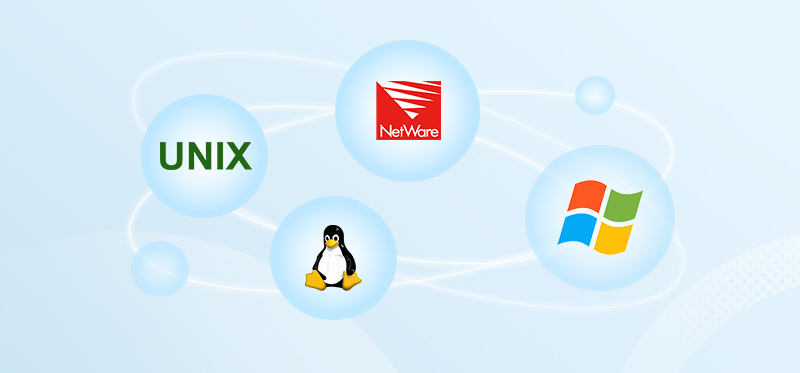
GUILLERMO DE JESÚS VAZQUEZ OLIVA (202164895)

DANA CONCEPCIÓN VICENTE DE LOS SANTOS (202165637)

**CICLO ESCOLAR:** AGOSTO – DICIEMBRE 2023

**FECHA:** MARTES 04 DE OCTUBRE DEL 2023





**Cuestionario 4**

1. Dado los siguientes diagramas de grafos de precedencia, determine lo siguiente:

a) Definir cuantos semáforos necesita cada grafo

b) Obtener la notación semáforo

c) Código en C

1-

S1

S2

S3

S5

S6

1. En este caso son necesarios 6 semáforos

S4

b)

Diagrama

Descripción generada automáticamente

c)

**Imagen que contiene Texto

Descripción generada automáticamente Texto, Carta

Descripción generada automáticamente**

**Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente**

**Salida del programa:**

Texto

Descripción generada automáticamente

S1

S2

S3

S4

S5

S6

2-

1. En este caso se usan 6 semáforos
2. Notación semáforo: Diagrama

   Descripción generada automáticamente

Texto

Descripción generada automáticamente Texto

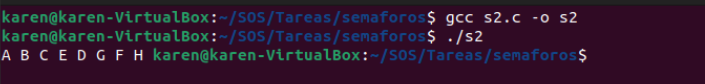
Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Salida:



3-

A

G

B

D

C

F

E

1. Se usaron 6 semáforos

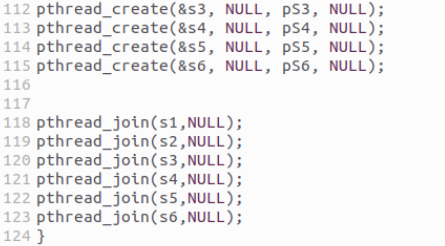
Diagrama

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza bajaTexto

Descripción generada automáticamente



**Salida del programa:**

Texto

Descripción generada automáticamente

4. Dadas las condiciones obtener la notación semáforo para realizar el grafo y el programa en C usando semáforos para sincronizar y controlar la exclusión mutua.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Condiciones:**

P3 no puede ejecutar “e” sino se ha procesado P1 y d de P2.

P4 no puede imprimir “h” sino se ha procesado b de P1 y d de P2

**Notación semáforo:**

P3

P2

P1

P4

down. d

g

down.mutex

up.mutex

up.g

down. g

down.b

down.d

h

down.mutex

up.mutex

down. d

e

down.mutex

up.mutex

up.e

down. e

f

down.mutex

up.mutex

up.f

down. b

c

down.mutex

up.mutex

up.c

down. c

d

down.mutex

up.mutex

up.dup.h

down.mutex

a

up.mutex

up.a

down. a

b

down.mutex

up.mutex

up.bup.h

Grafo:

P4

P3

P1

P2

g

h

f

e

d

c

b

a

mutex

mutex

mutex

mutex

mutex

Sem2

Sem1

Código:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

sem\_t mutex;

sem\_t sem1;

sem\_t sem2;

int main() {

// Inicialización de semáforos

sem\_init(&mutex, 0, 1);

sem\_init(&sem1, 0, 0);

sem\_init(&sem2, 0, 0);

// Proceso P1

sem\_wait(&mutex);

// Imprime "a"

printf("a\n");

sem\_post(&mutex);

sem\_post(&sem1);

// Proceso P2

sem\_wait(&sem1);

// Imprime "b"

printf("b\n");

sem\_post(&mutex);

sem\_post(&sem2);

// Proceso P3

sem\_wait(&mutex);

// Imprime "c"

printf("c\n");

sem\_post(&mutex);

sem\_post(&sem1);

// Proceso P4

sem\_wait(&sem1);

// Imprime "d"

printf("d\n");

sem\_wait(&mutex);

// Imprime "e"

printf("e\n");

sem\_post(&mutex);

sem\_post(&sem2);

// Proceso P3

sem\_wait(&sem2);

// Imprime "f"

printf("f\n");

sem\_post(&mutex);

// Proceso P4

sem\_wait(&mutex);

// Imprime "g"

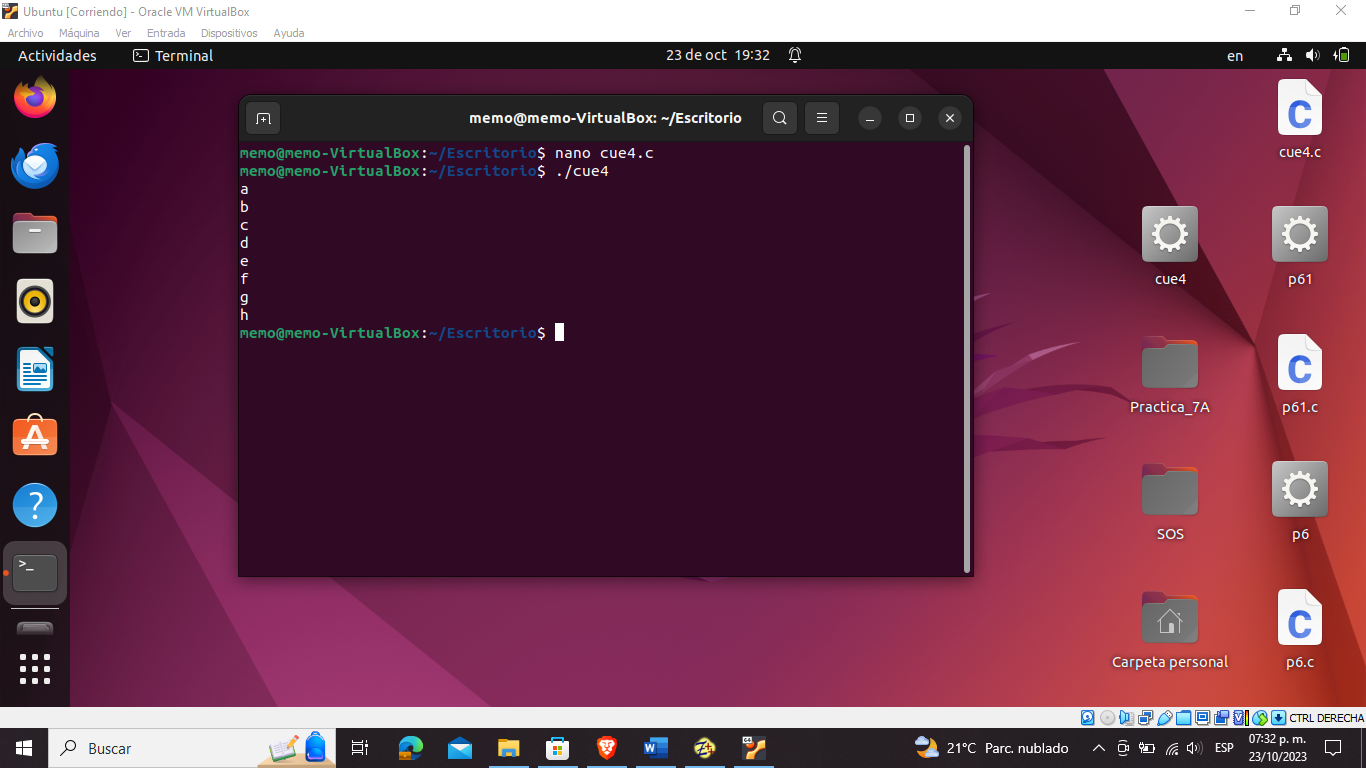
printf("g\n");

// Imprime "h"

printf("h\n");

sem\_post(&mutex);

return 0;



5. Dadas las condiciones obtener la notación semáforo para realizar el grafo y el programa en C usando semáforos para sincronizar y controlar la exclusión mutua, además calcular el valor leyendo los datos.

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

**Notación semáforo:**

P3

P2

P1

P4

down.c

S4

down.mutex

up.mutex

up.d

down.mutex

S1

up.mutex

up.a

down.a

down. d

down.c

S4

down.mutex

up.mutex

up.g

down. a

down. b

S3

down.mutex

up.mutex

up.c

S2

down.mutex

up.mutex

up.b

Grafo:

P4

P3

P2

P1

Sem\_4

Sem\_1

Sem\_2

S2

S1

Sem\_3

S3

Sem\_1

S4

**Código:**

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

sem\_t sem\_P1, sem\_P2, sem\_P3, sem\_P4;

int a, b, c, d, e, f, g, h, w, x, z;

void \*P1\_thread(void \*arg) {

// Realiza cálculos basados en datos leídos

c = a + b;

sem\_post(&sem\_P1); // Libera sem\_P1 para permitir que P2 continúe

pthread\_exit(NULL);

}

void \*P2\_thread(void \*arg) {

sem\_wait(&sem\_P1); // Espera a que P1 termine

// Realiza cálculos basados en datos leídos

e = c + f;

sem\_post(&sem\_P2); // Libera sem\_P2 para permitir que P3 continúe

pthread\_exit(NULL);

}

void \*P3\_thread(void \*arg) {

sem\_wait(&sem\_P2); // Espera a que P2 termine

// Realiza cálculos basados en datos leídos

w = e + g;

sem\_post(&sem\_P3); // Libera sem\_P3 para permitir que P4 continúe

pthread\_exit(NULL);

}

void \*P4\_thread(void \*arg) {

sem\_wait(&sem\_P3); // Espera a que P3 termine

// Realiza cálculos basados en datos leídos

x = w + h;

z = x; // Establece z igual a x

pthread\_exit(NULL);

}

int main() {

// Inicializa semáforos

sem\_init(&sem\_P1, 0, 0);

sem\_init(&sem\_P2, 0, 0);

sem\_init(&sem\_P3, 0, 0);

// Lee los datos necesarios

printf("Ingrese el valor de a: ");

scanf("%d", &a);

printf("Ingrese el valor de b: ");

scanf("%d", &b);

printf("Ingrese el valor de f: ");

scanf("%d", &f);

printf("Ingrese el valor de g: ");

scanf("%d", &g);

printf("Ingrese el valor de h: ");

scanf("%d", &h);

pthread\_t thread\_P1, thread\_P2, thread\_P3, thread\_P4;

// Crea los hilos para los procesos P1, P2, P3 y P4

pthread\_create(&thread\_P1, NULL, P1\_thread, NULL);

pthread\_create(&thread\_P2, NULL, P2\_thread, NULL);

pthread\_create(&thread\_P3, NULL, P3\_thread, NULL);

pthread\_create(&thread\_P4, NULL, P4\_thread, NULL);

// Espera a que los hilos terminen

pthread\_join(thread\_P1, NULL);

pthread\_join(thread\_P2, NULL);

pthread\_join(thread\_P3, NULL);

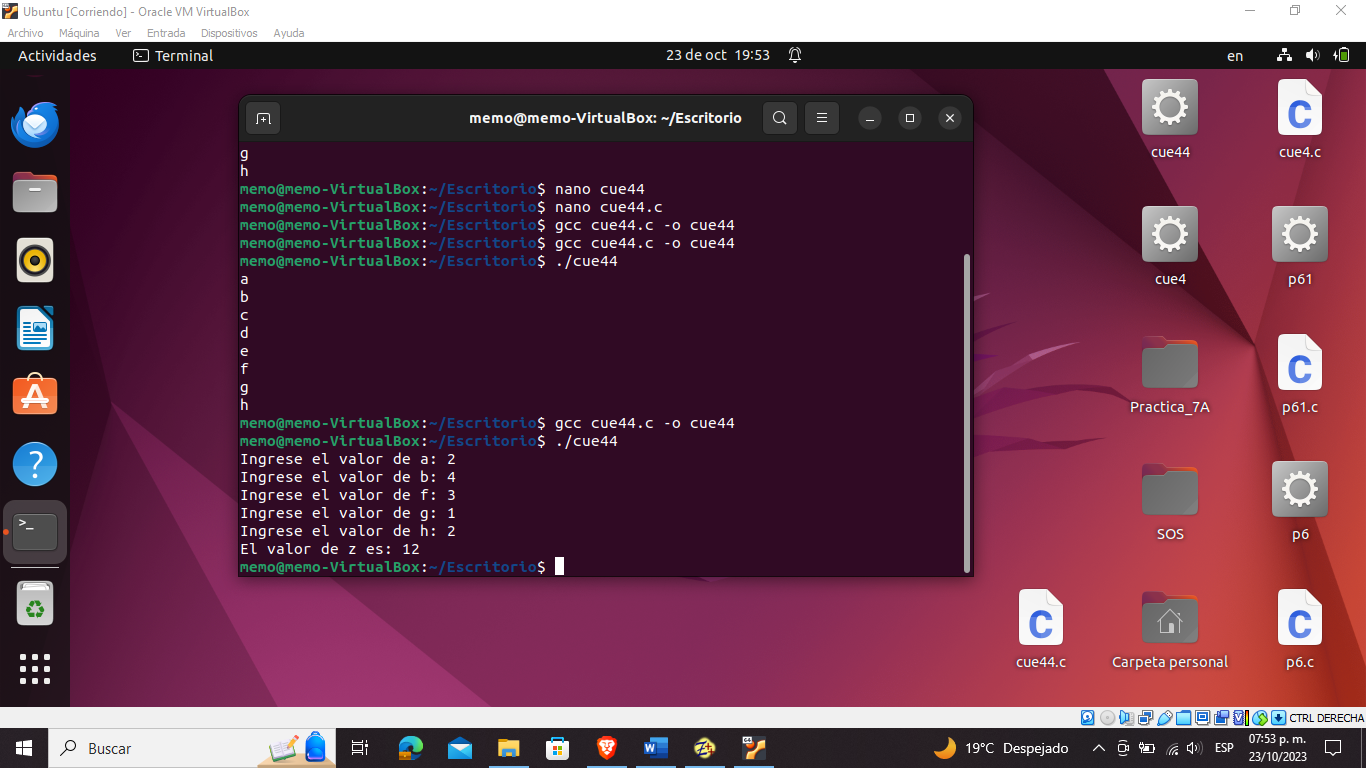
pthread\_join(thread\_P4, NULL);

// Imprime el valor calculado de z

printf("El valor de z es: %d\n", z);

return 0;

}



Responda:

1. ¿Cuáles son los dos problemas principales inherentes a la programación concurrente?

* Exclusión mutua: cuando dos procesos se ejecutan concurrentemente y comparten las mismas variables, es posible que haya sobreescritura del valor de las mismas dando lugar a resultados no deseables.
* Sincronización y comunicación: hay situaciones en las que un recurso compartido por varios procesos se encuentra en un estado en el que un proceso no puede hacer una determinada acción con el hasta que no cambie su estado.

1. ¿Qué es una sección critica?

Es la parte del código de un proceso concurrente donde se utiliza un recurso compartido.

1. ¿Qué es un hilo y cuáles son sus características?

Un hilo es una secuencia de código en ejecución dentro del contexto de un proceso. Los hilos no pueden ejecutarse ellos solos. Requieren la supervisión de un proceso padre para correr.

Características:

* Estado.
* Contexto del procesador: punto en el que estamos ejecutando, la instrucción concretamente en la que nos hallamos. Es útil a la hora de reanudar un hilo que fue interrumpido con anterioridad, puesto que, al guardar el contexto, guardamos la última instrucción que ejecutamos, y así podemos conocer por donde tenemos que continuar la ejecución del hilo.
* Pila de ejecución donde se irá metiendo y sacando instrucciones. (Lugar donde almacenaremos las instrucciones que van a ser ejecutadas).
* Espacio de almacenamiento estático donde almacenará las variables.
* Acceso a los recursos de la tarea, que son compartidos por todos los hilos de la tarea.

1. ¿Cuál es la diferencia principal entre un hilo y un proceso?

Tienen diferentes formas de administrar los recursos en el sistema operativo.

1. ¿Cuáles son los dos niveles de hilos que existen?

Aquellos que nosotros usaremos para programar y que pueden crearse desde lenguajes de programación y aquellos otros hilos del propio SO que sirven para dar soporte a nuestros hilos de usuario y que son los hilos del sistema.

1. ¿Qué son los hilos Poxis?

Permiten la ejecución concurrente de varias secuencias de instrucciones asociadas a funciones dentro de un mismo proceso (hilo principal).

1. ¿Qué es un semáforo y para que se utiliza?

Son mecanismos que permiten sincronizar procesos para prevenir colisiones que se producen cuando dos o más procesos solicitan información del uso de un recurso que deben compartir.

Sirven para implantar exclusión mutua e implantar sincronización entre procesos.

1. Tipos de semáforos en Linux.

Semáforos POSIX:

* Semáforos nombrados: se puede utilizar por el proceso que lo crea y por cualquier otro aunque no tenga relación con el creador.
* Semáforos no nombrados: se puede utilizar por el proceso que lo crea (threads). El uso desde otros procesos precisa de una región de memoria compartida.

1. ¿Cuál es la estructura del semáforo?

* El valor del semáforo.
* El identificador del ultimo proceso que manipulo el semáforo.
* El numero de procesos que hay esperando a que el valor del semáforo se incremente.
* El numero de procesos que hay esperando que el semáforo tome el valor de 0.

1. ¿Cuáles son las dos operaciones básicas del semáforo?

Down(p): verifica si el valor de un semáforo es mayor que cero, en cuyo caso decrementa el valor. Si el valor es cero entonces el proceso se va a dormir. La verificación, la modificación y la posibilidad de irse a dormir se realiza en conjunto, como una sola e indivisible acción atómica.

Up(v): incrementa el valor del semáforo.

P (espera o wait).

V (señal o signal).

Referencias:

[Microsoft PowerPoint - HilosPosix (unam.mx)](http://lcomp89.fi-b.unam.mx/licad/assets/ProgramacionMultitareasHilos/HilosPosix.pdf#:~:text=Hilos%20POSIX%20Permiten%20la%20ejecuci%C3%B3n%20concurrente%20de%20varias,y%20ficheros%20que%20tuviera%20abierto%20el%20hilo%20principal.)

[ConceptoHilos (uam.mx)](http://aisii.azc.uam.mx/areyes/archivos/licenciatura/sd/U2/ConceptoHilos.pdf)

[Procesos e hilos • Sistemas Operativos • WebProgramacion Consultoría Informática](https://webprogramacion.com/procesos-e-hilos/)

[Microsoft Word - capitulo2.Procesos e Hilos.doc (buap.mx)](http://mceron.cs.buap.mx/cap2_dis.pdf#:~:text=Podemos%20hablar%20de%20dos%20niveles%20de%20hilos%3A%20aquellos,los%20hilos%20del%20sistema%20seg%C3%BAn%20la%20Figura%205.)