

cuestionamientos en las hojas qualitas respuestas en blanco no se Parcial (2 pts.), e Incompleta (0	ue se le entregan marcando toman en cuenta). <i>Las últim</i>	salirse de ellas (por ejemplo) una única opción. Se utilizar as tres preguntas se evaluarán de estas últimas tres preguntas, sino	á factor de corrección 4 a 1 e manera Completa (4 pts.),
indicado.			
$igcircle{0} igcircle{0} igcolumn{0} 0 igcolumn{0} igcolumn{0} 0 igcolumn{$	$\begin{array}{c} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{array}$	← Marque su RUT sin co cador (el número después de escriba sus nombres y apellie Nombre(s) y apellido(s):	el guión), y
$ \begin{array}{c} $	6 6 6 6 6 $7 7 7 7$ $8 8 8 8$ $9 9 9 9$		
 ¿Qué es una variable de condic Varialbe de sincronización que implementa yield Variable de sincronización que permite la exclusión mutua En la creación del proceso del de implementa de la implementa de la inicial de la in	 ○ Variable de sincronización que permite la espera eficiente de un hilo ○ Objeto de sincronización que implementa yield en modo usuario programa prog, ¿qué paso es 	7. ¿Qué es el multi-hilado prevent Es cuando los hilos que se ejecutan no pueden descalendarizarse Es cuando los hilos no usan el procesador hasta que se les entrega indefi- nidamente	tivo? Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos entregan el procesador voluntariamente, no son interrumpidos
el siguiente después de inicializar e Informar al calendariza- dor que el programa se puede ejecutar Cargar prog en la memo- ria ¿Cuál de las siguientes es una g	 Copiar los argumentos en la memoria Inicializar el hardware para que se ejecute el incio del programa 	8. ¿Cuál es el principal problema No podemos tener varios hilos por proceso El bloqueo del proceso, bloquea todos los hilos	de los hilos a nivel de usuario? No podemos calendarizar hilos dentro del proceso No podemos tener varios procesos por hilo
 Copiar a la memoria el código del programa del proceso padre Copiar el proceso padre completamente Tipo de kernel en el que la func 	 Crear una copia del proceso padre, pero no puede ser confiado igual que él Copiar el proceso padre con privilegios distintos 	 Qué es un sistema operativo? No es software, pero se encarga de administrar recursos y usuarios Es una capa de software que administra recursos y usuarios 	 Es un software que administra recursos Es un software que administra usuarios
de él, y el resto se encuentra a nive		10. ¿Cuál es el primer paso en la	creación del proceso del pro-
Microkernel Monolítico	◯ Híbrido◯ Macrokernel	grama prog? Inicializar la memoria Copiar prog a la memo-	O Inicializar el hardware para que se ejecute el
5. ¿Qué hace la función wait de u Duerme el hilo atómica- mente mientras despier-	Mueve el hilo a la lista de espera atómicamente	ria	proceso desde el inicio Crear e inicializar el PCB en el kernel
ta al siguiente Atómicamente libera los seguros y mueve el hilo a la lista de espera	Libera el seguro global, y duerme al hilo	11. ¿Cuál de los siguientes evento modo kernel a usuario?Continuar después de una interrupción	Creación de nuevo proce-
6. ¿Cuál de los siguientes elemen hilos?	tos no es compartido por los	Llamada I/O a un dispositivo	Cambiar a un proceso di- ferente
○ Heap○ Código	StackVariables globales		

Para revisión, preferentemente imprimir documentos compilados desde automultiple-choice.



 12. ¿Qué hace la función yield d Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished) 	e un hilo? Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (waiting) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready)	 21. ¿Cuándo se da una condición Varios hilos se ejecutan simultáneamente Existe una sección crítica ca 22. Región de la memoria reservada mantener el estado de las variables 	 El estado de la ejecución de un programa depende del intercalado de diferentes hilos Varios hilos se pueden intercalar en su ejecución simultánea a por el sistema operativo para
 13. En la creación del proceso del el siguiente después de cargar progono Copiar los argumentos en la memoria Informar al calendarizador que el programa se puede ejecutar 		funciones. Heap RAM 23. Modo de ejecución del proces guna verificación. Modo inseguro	Modo de usuario
14. Modo de ejecución del proceinstrucción antes de ejecutarla. Modo dual Modo seguro	sador en el que verifica cada Modo de usuario Modo kernel	 Modo kernel 24. ¿Qué es una operación atómic Es una operación indivisible que no puede ser dividida 	 Modo dual ea? Es una operación del kernel que no puede ser dividida
15. Tipo de kernel en el que la méste se encuentra dentro de él. Microkernel Macrokernel	nayoría de la funcionalidad de Monolítico Híbrido	Vidida Es una operación de hardware que no puede ser dividida 25. ¿Qué son las direcciones de m	Es una operación que deshabilita las interrupciones
 16. ¿Qué característica del sistem independientemente del hardware	en la computadora? ☐ Ilusionista ☐ Virtualización na?	Capa de indirección que le da flexibilidad al sistema operativo para administrar la memoria Memoria que reserva el sistema operativo para el uso de máquinas virtuales	 Memoria adicional que crea el sistema operativo para engañar a las aplicaciones Capa adicional que utiliza el sistema operativo en modo kernel
 Funciones del kernel que permiten al usuario accesar a recursos restringidos Código del kernel que ejecuta código de usuario 	 Llamada que hace el kernel para realizar una instrucción Funciones de hardware que llama el usuario 	26. ¿Qué es la ejecución de una apgidos? Hilo Proceso	Olicación con permisos restrin- O Aplicación O Programa
18. ¿Cuál de los siguientes elemen Estado del hilo Puntero al stack frame 19. ¿Qué es la exclusión mutua? Propiedad donde los hi-	Registros salvados Variables locales Propiedad de la concu-	27. ¿Qué es un pipe? Es un buffer del kernel entre dos descriptores de archivos Es un buffer de usuario entre dos descriptores de archivos	 Es un buffer del kernel entre dos procesos Es un buffer de usuario entre dos procesos
los se bloquean mutua- mente Objeto de sincronización que bloquea los hilos 20. Es una señal síncrona al proc	rrencia para mantener los hilos excluidos Propiedad en la que solamente un hilo puede accesar simultáneamente esador que indica que ocurrió	28. ¿Donde almacena el sistema op sobre un proceso en particular? O Heap O Stack	Operativo la toda la información
un evento que requiere su atención Interrupción Trampa		 29. ¿Cuál de las siguientes afirmades spin wait? Es justificable en monoprocesadores No se debe de usar nunca 	,



Dado el siguiente código

}	
30. ¿Cuál de los siguientes no es x al término de los procesos del pro-	
520	○ 10○ 15
31. ¿Cuántas copias distintas de l cución del programa anterior?	a variable \mathbf{x} existen en la eje-
\bigcirc 4 \bigcirc 2	$\bigcirc 1 \\ \bigcirc 3$
32. ¿Qué es un hilo? ○ Varias secuencias de ejecución que pueden ser calendarizadas dependientemente ○ Varias secuencias de ejecución que pueden ser calendarizadas independientemente 33. ¿Cuál de las siguientes no es criptor de archivo de UNIX? ○ Se abren antes de usar ○ Se cierran completamente después de usar	 Una secuencia de ejecución que es calendarizada dependientemente Una secuencia de ejecución que puede ser calendarizada independientemente una característica de un dessona de le secuención que puede ser calendarizada independientemente Se leen a través de un buffer Se escriben a través de bytes
34. Llamada del sistema de UNIX ciones el comunicarse entre sí para resumirlas. \(\begin{align*} \text{wait} \\ \end{align*} \text{signal} \end{align*}	
35. ¿Cuál es el estado de un hilo e tarse pero no está en el procesador Ejecutandose (running) Inicializado (init)	

36. ¿Cuántos procesos son creados durante la ejecución del si $guiente\ programa?$

	<pre>main(int argc, char** argv){ forkthem(5);</pre>
	}
	<pre>void forkthem(int n){ if (n > 0){ fork(); fork(); }</pre>
	forkthem(n-1); }
	}
5 30	$ \bigcirc 16 $ $ \bigcirc 4 $

37. ¿Cuál de las siguientes \mathbf{no} es u	una característica de un hilo?
Comparte el código del	O Tiene un stack propio
programa	O Tiene un segmento de
O Tiene un contador de	datos propio
programa propio	
38. Es una señal asíncrona al proce	esador que indica que ocurrió
un evento que requiere su atención.	
O Interrupción	○ Trampa
O Señal de software	O Señal de I/O
39. ¿Qué es concurrencia?	
O Realizar una actividad	O Realizar múltiples activi-
sin la interrupción de	dades sin la interrupción
otra	de otras
Realizar múltiples actividades simultáneamente	Realizar múltiples actividades una después de la
dades simultaneamente	otra
40. Región de la memoria reserva	ada por el sistema operativo
para alojar estructuras de datos qu	e el proceso pueda necesitar.
O Heap	○ Stack
\bigcirc RAM	○ Swap



```
1
   char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
10
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcirc I \bigcirc P \bigcirc C$ manera segura en la lista circular.

manora sogura en la neva en curation.	01 01 00

43. Reescriba la función Queue::remove para que elimine de

manera segura en la lista circular.	\bigcirc I \bigcirc P \bigcirc C





cuestionamientos en las hojas qu (las respuestas en blanco no se t	ue se le entregan marcando coman en cuenta). <i>Las últim</i>	salirse de ellas (por ejemplo •) una única opción. Se utilizar as tres preguntas se evaluarán de estas últimas tres preguntas, sino	á factor de corrección 4 a 1 e manera Completa (4 pts.),
$\begin{array}{c} \bigcirc 0\bigcirc 0$	$\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 $	← Marque su RUT sin co cador (el número después de escriba sus nombres y apellio	el guión), y
$ \begin{array}{c ccccc} & & & & & & & & & & & & & & & & & & &$	$egin{array}{cccccccccccccccccccccccccccccccccccc$	Nombre(s) y apellido(s):	
1. ¿Qué es concurrencia? Realizar múltiples actividades sin la interrupción	Realizar una actividad sin la interrupción de	7. Es una señal asíncrona al proc un evento que requiere su atención	
de otras Realizar múltiples activi-	otra (Realizar múltiples activi-	Señal de software	○ Señal de I/O
dades una después de la otra 2. ¿Qué es el multi-hilado prevent Es cuando los hilos no usan el procesador hasta que se les entrega indefinidamente Es cuando los hilos que se ejecutan no pueden descalendarizarse 3. Tipo de kernel en el que la func de él, y el resto se encuentra a nive Microkernel Monolítico 4. Modo de ejecución del proces instrucción antes de ejecutarla.	Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos entregan el procesador voluntariamente, no son interrumpidos ionalidad mínima está dentro el de usuario. Macrokernel Híbrido	 8. ¿Qué son las direcciones de me le da flexibilidad al sistema operativo para administrar la memoria Capa adicional que utiliza el sistema operativo en modo kernel 9. ¿Cuándo se da una condición de un programa depende del intercalado de diferentes hilos Varios hilos se pueden intercalar en su ejecución simultánea 10. Modo de ejecución del proces 	 Memoria adicional que crea el sistema operativo para engañar a las aplicaciones Memoria que reserva el sistema operativo para el uso de máquinas virtuales Le carrera? Varios hilos se ejecutan simultáneamente Existe una sección crítica
Modo de usuario Modo kernel	Modo seguroModo dual	guna verificación. Modo inseguro	O Modo dual
5. ¿Cuál de los siguientes elementables?	tos no es compartido por los	() Modo kernel 11. ¿Cuál de las siguientes afirma	Modo de usuario ciones es verdadera respecto a
○ Stack ○ Variables globales	Código Heap	spin wait? Produce deadlocks Es justificable en mono-	Es justificable en multi- procesadoresNo se debe de usar nunca
6. ¿Qué es la ejecución de una ap gidos?Proceso	O Programa	procesadores	Tro se debe de usar nunca
Hilo	Aplicación		



12. ¿Qué es una llamada de sistema? Código del kernel que ejecuta código de usuario trucción Funciones de hardware que llama el usuario Código del kernel que ejecuta código de usuario permiten al usuario accesar a recursos restringi-	20. ¿Cuál es el principal problema de los hilos a nivel de usuario? No podemos calendarizar hilos dentro del proceso El bloqueo del proceso, bloquea todos los hilos No podemos tener varios procesos por hilo No podemos tener varios hilos por proceso
dos	Dado el siguiente código
13. ¿Qué es un sistema operativo? Es un software que administra recursos Es una capa de software que administra recursos y usuarios Es un software que administra recursos y usuarios Es un software que administra usuarios	<pre>main(int argc, char** argv){ int child == fork(); int x = 5; if (child == 0) { x += 5; } else {</pre>
14. ¿Qué hace la función yield de un hilo?	child = fork();
 ○ Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (waiting) ○ Entrega voluntariamente el resto del ciclo en el procesador, y cambia el ○ Entrega voluntariamente el resto del ciclo en el procesador, y cambia el ○ Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready) 	x += 10; if (child) {
estado a terminado (fi-	\bigcirc 20 \bigcirc 5
nished)	\bigcirc 20
15. ¿Qué característica del sistema operativo permite que opere independientemente del hardware en la computadora? O Ilusionista O Compartimiento de recursos O Virtualización O Portabilidad	22. ¿Cuántas copias distintas de la variable x existen en la ejecución del programa anterior? 2
	23. Es una señal síncrona al procesador que indica que ocurrió
16. ¿Qué hace la función wait de una variable de condición? ○ Libera el seguro global, y duerme al hilo seguros y mueve el hilo a la lista de espera mente mientras despierta al siguiente Atómicamente una variable de condición? ○ Atómicamente libera los seguros y mueve el hilo a la lista de espera atómicamente	un evento que requiere su atención. O Interrupción O Trampa O Señal de software O Señal de I/O 24. ¿Cuántos procesos son creados durante la ejecución del siguiente programa?
17. ¿Cuál es el estado de un hilo en el que está listo para ejecu-	
tarse pero no está en el procesador? O Inicializado (init) O Esperando (wait) O Listo (ready)	<pre>main(int argc, char** argv){ forkthem(5); } void forkthem(int n){ if (n > 0){</pre>
18. ¿Qué es una operación atómica?	fork();
Es una operación que deshabilita las interrupciones Es una operación del kernel que no puede ser dividida	forkthem(n-1); } 0 4
Es una operación de hardware que no puede ser dividida Es una operación indivisible que no puede ser dividida	30525. En la creación del proceso del programa prog, ¿qué paso es
19. ¿Qué es un hilo? Varias secuencias de ejecución que pueden ser calendarizadas independientemente Una secuencia de ejecución que es calendarizada Una secuencia de ejecución que es calendarizada Una secuencia de ejecución que puede ser calendarizada	el siguiente después de inicializar el espacio de memoria? Inicializar el hardware para que se ejecute el incio del programa Cargar prog en la memoria Cargar prog en la memoria Cargar prog en la memoria Difformar al calendarizador que el programa se puede ejecutar
dependientemente darizada independiente-	

mente



los hilos excluidos

 26. ¿Qué es un pipe? Es un buffer del kernel entre dos procesos Es un buffer de usuario entre dos descriptores de archivos 27. En la creación del proceso del el siguiente después de cargar progonicializar el PCB en el kernel Copiar los argumentos en la memoria 		 35. ¿Cuál de las siguientes no es Tiene un stack propio Comparte el código del programa 36. ¿Cuál de los siguientes evento modo kernel a usuario? Creación de nuevo proceso Continuar después de una interrupción 37. ¿Cuál de las siguientes es una Copiar a la memoria el 	 Tiene un segmento de datos propio Tiene un contador de programa propio s no genera una transición de Llamada I/O a un dispositivo Cambiar a un proceso diferente
 28. ¿Donde almacena el sistema op sobre un proceso en particular? Stack Memoria 29. ¿Qué es una variable de conde 	Ocerativo la toda la información Heap Process control block (PCB)	código del programa del proceso padre Copiar el proceso padre con privilegios distintos 38. Tipo de kernel en el que la néste se encuentra dentro de él.	completamente Crear una copia del proceso padre, pero no puede ser confiado igual que él nayoría de la funcionalidad de
 Varialbe de sincronización que implementa yield Variable de sincronización que permite la espera eficiente de un hilo 	 Objeto de sincronización que implementa yield en modo usuario Variable de sincronización que permite la exclusión mutua 	 	 Monolítico Híbrido una característica de un des- Se leen a través de un buffer
 30. ¿Cuál es el primer paso en la grama prog? Inicializar el hardware para que se ejecute el proceso desde el inicio Crear e inicializar el PCB en el kernel 	Creación del proceso del pro- Copiar prog a la memoria Inicializar la memoria	Se abren antes de usar 40. Región de la memoria reservara alojar estructuras de datos q RAM Stack	
31. Llamada del sistema de UNI ciones el comunicarse entre sí para resumirlas. O signal O fork			
32. Región de la memoria reservado mantener el estado de las variables funciones.RAMSwap	s locales durante la llamada a O Stack		
33. ¿Cuál de los siguientes elemen Registros salvados	Estado del hilo		
 Variables locales 34. ¿Qué es la exclusión mutua? Propiedad en la que solamente un hilo puede accesar simultáneamente Propiedad de la concurrencia para mantener 	 Objeto de sincronización que bloquea los hilos Propiedad donde los hilos se bloquean mutuamente 		



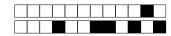
```
1
   char bar(){
2
     int b = 2;
3
      return 'a';
4 }
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
10
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcirc I \bigcirc P \bigcirc C$ manera segura en la lista circular.

г	
1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
1	
- 1	
- 1	
- 1	
- 1	
- 1	
- 1	
1	
- 1	
- 1	
- 1	
- 1	
1	
ы	
VI	
1	
ı	
ı	
-1	
1	
-1	
-1	
1	
1	
-1	
1	
1	
ı	
ı	
-1	
-1	
1	
1	
1	
1	
-1	
ы	
All	
1	
1	

43. Reescriba la función Queue::remove para que elimine de nora cogura on la lista circular $\bigcirc I \bigcirc D \bigcirc C$

nanera segura en la lista circular.	OI OP	\bigcirc C



 ${\rm de\ otras}$

Solemne 1 SO (CIT 2003-1)

cuestionamientos en las hojas que se le entregan marcando (las respuestas en blanco no se toman en cuenta). <i>Las últim</i>	s salirse de ellas (por ejemplo ●). Responda a los siguientes o una única opción. Se utilizará factor de corrección 4 a la as tres preguntas se evaluarán de manera Completa (4 pts.) e estas últimas tres preguntas, sino que responda en el espacio
$ \bigcirc 0 \bigcirc$	← Marque su RUT sin código verificador (el número después del guión), y escriba sus nombres y apellidos abajo. Nombre(s) y apellido(s):
1. ¿Qué son las direcciones de memoria virtuales? Capa adicional que utiliza el sistema operativo en modo kernel Memoria que reserva el sistema operativo para el uso de máquinas virtuales Capa de indirección que le da flexibilidad al sistema operativo para administrar la memoria Memoria adicional que crea el sistema operativo para engañar a las aplicaciones	6. Modo de ejecución del procesador donde no se ejecuta ninguna verificación. \(\begin{align*} \text{Modo inseguro} & \text{Modo kernel} \\ \text{Modo dual} \end{align*} 7. ¿Cuál de las siguientes no es una característica de un hilo? \(\begin{align*} \text{Tiene} & \text{un contador de} & \text{Tiene un stack propio} \\ \text{propio} & \text{Tiene un segmento de datos propio} \\ \text{Comparte el código del} \end{align*}
2. En la creación del proceso del programa prog, ¿qué paso es el siguiente después de cargar prog en la memoria? O Informar al calendarizador que el programa se puede ejecutar O Inicializar el hardware para que se ejecute el incio del programa 3. Región de la memoria reservada por el sistema operativo para	programa 8. ¿Cuál de las siguientes afirmaciones es verdadera respecto a spin wait? O No se debe de usar nunca Es justificable en monoprocesadores Produce deadlocks 9. ¿Cuál de los siguientes elementos no pertenece al TCB?
Alojar estructuras de datos que el proceso pueda necesitar. Swap Heap Stack	 ○ Puntero al stack frame ○ Registros salvados ○ Variables locales 10. ¿Qué es la exclusión mutua? ○ Propiedad en la que sola- ○ Propiedad de la concu-
 ¿Qué es una operación atómica? Es una operación que deshabilita las interrupciones Es una operación de hardware que no puede Es una operación de les una operación del kernel que no puede 	mente un hilo puede accesar simultáneamente Objeto de sincronización que bloquea los hilos Trencia para mantener los hilos excluidos Propiedad donde los hilos se bloquean mutuamente
ser dividida vidida 5. ¿Qué es concurrencia? Realizar múltiples actividades simultáneamente Realizar múltiples actividades sin la interrupción de otra Realizar múltiples actividades sin la interrupción Realizar múltiples actividades sin la interrupción	11. ¿Cuál de los siguientes eventos no genera una transición de modo kernel a usuario? Continuar después de Llamada I/O a un dispouna interrupción sitivo Cambiar a un proceso diferente so

dades una después de la

otra



 12. ¿Qué hace la función yield of Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (waiting) 	de un hilo? Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready) Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting)	20. ¿Qué es un pipe? Es un buffer de usuario entre dos procesos Es un buffer del kernel entre dos descriptores de archivos 21. Región de la memoria reservad mantener el estado de las variable funciones. Stack	es locales durante la llamada a O Swap
13. Llamada del sistema de UNI ciones el comunicarse entre sí par resumirlas.		 RAM ¿Cuál de las siguientes no escriptor de archivo de UNIX?	Heap una característica de un des-
exec	○ signal	Se cierran completamen-	O Se leen a través de un
○ wait	O fork	te después de usar	buffer
14. ¿Cuál es el estado de un hilo	en el que está listo para ejecu-	O Se escriben a través de bytes	O Se abren antes de usar
tarse pero no está en el procesado	r?	23. ¿Cuál de los siguientes eleme	entos no es compartido por los
○ Ejecutandose (running)	○ Inicializado (init)	hilos?	,
C Esperando (wait)	\bigcirc Listo $(ready)$	O Heap	○ Stack
15. ¿Qué es un sistema operativo	o? _	○ Código	O Variables globales
Es un software que administra recursos Es un software que administra recursos	Es una capa de software que administra recursos y usuarios	24. ¿Qué es la ejecución de una a gidos?	plicación con permisos restrin-
nistra usuarios	No es software, pero se	O Hilo	O Programa
	encarga de administrar recursos y usuarios	Aplicación	O Proceso
16. Tipo de kernel en el que la r	navoría de la funcionalidad de	25. ¿Cuándo se da una condición Varios hilos se ejecutan	de carrera? Existe una sección críti-
éste se encuentra dentro de él.	najona de la rancionazada de	simultáneamente	ca
○ Macrokernel○ Monolítico	◯ Híbrido◯ Microkernel	Varios hilos se pueden in- tercalar en su ejecución simultánea	El estado de la ejecución de un programa depen- de del intercalado de di-
17. Es una señal síncrona al pro- un evento que requiere su atenció:			ferentes hilos
	Señal de software	26. ¿Cuántos procesos son cread guiente programa?	los durante la ejecución del si-
○ Interrupción○ Señal de I/O	Trampa	guiente programa.	
		main(int argc,	char** argv){
18. ¿Qué hace la función wait de		<pre>forkthem(5); }</pre>	
Atómicamente libera los seguros y mueve el hilo a	Mueve el hilo a la lista de espera atómicamente	<pre>void forkthem(i: if (n > 0){</pre>	nt n){
la lista de espera	C Libera el seguro global, y	fork();	
O Duerme el hilo atómica-	duerme al hilo	forkthem(n- }	1);
mente mientras despier- ta al siguiente		}	
		\bigcirc 5	\bigcirc 30
19. ¿Qué es un hilo? O Una secuencia de ejecu-	O Una secuencia de ejecu-	\bigcirc 4	\bigcirc 16
ción que es calendarizada	ción que puede ser calen-	27. Modo de ejecución del proc	asador an al qua varifica cada
dependientemente	darizada independiente-	instrucción antes de ejecutarla.	coador en er que vermea cada
O Varias secuencias de	mente	Modo de usuario	Modo kernel
ejecución que pue- den ser calendarizadas dependientemente	Varias secuencias de eje- cución que pueden ser calendarizadas indepen-	Modo seguro	Modo dual

dientemente



zar hilos dentro del pro-

ceso

00 .0 / 1 1:1:1 1 2	or .O./
28. ¿Qué es el multi-hilado preventivo? Es cuando los hilos que se ejecutan no pueden descalendarizarse Es cuando los hilos no usan el procesador hasta que se les entrega indefinidamente Es cuando los hilos no terrumpidos Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente	35. ¿Qué es una variable de condición? Variable de sincronización que permite la espera eficiente de un hilo yield Variable de sincronización que permite la exclusión mutua en modo usuario 36. ¿Qué característica del sistema operativo permite que opere independientemente del hardware en la computadora?
Dado el siguiente código	Ilusionista Portabilidad
<pre>main(int argc, char** argv){ int child == fork();</pre>	Compartimiento de re- cursos
int x = 5; if (child == 0) {	37. En la creación del proceso del programa prog, ¿qué paso es el siguiente después de inicializar el espacio de memoria?
<pre>x += 5; } else { child = fork(); x += 10; if (child) {</pre>	O Informar al calendarizador que el programa se puede ejecutar O Copiar los argumentos en la memoria O Cargar prog en la memo-
x+=5; }	O Inicializar el hardware ria para que se ejecute el incio del programa
}	38. Es una señal asíncrona al procesador que indica que ocurrió
29. ¿Cuál de los siguientes no es un valor que toma la variable	un evento que requiere su atención.
x al término de los procesos del programa anterior?	O Interrupción O Señal de software
$ \begin{array}{ccc} $	○ Señal de I/O ○ Trampa
30. ¿Cuántas copias distintas de la variable $\mathbf x$ existen en la eje-	${\bf 39.}$ Tipo de kernel en el que la funcionalidad mínima está dentro de él, y el resto se encuentra a nivel de usuario.
cución del programa anterior?	O Monolítico O Microkernel
\bigcirc 3 \bigcirc 1 \bigcirc 2	Macrokernel Híbrido
	40. ¿Cuál es el primer paso en la creación del proceso del pro-
31. ¿Donde almacena el sistema operativo la toda la información sobre un proceso en particular?	grama prog? Crear e inicializar el Copiar prog a la memo-
○ Stack ○ Process control block	PCB en el kernel ria
○ Heap (PCB) ○ Memoria	() Inicializar el hardware () Inicializar la memoria para que se ejecute el proceso desde el inicio
32. ¿Qué es una llamada de sistema?	processo desde el inicio
O Llamada que hace el kernel para realizar una inseque llama el usuario	
trucción	
Código del kernel que ejecuta código de usuario esar a recursos restringidos	
33. ¿Cuál de las siguientes es una característica de fork?	
Copiar el proceso padre completamente Copiar el proceso padre con privilegios distintos	
Copiar a la memoria el código del programa del proceso padre Crear una copia del proceso padre, pero no puede ser confiado igual que él	
34. $_i$ Cuál es el principal problema de los hilos a nivel de usuario?	
○ El bloqueo del proceso, ○ No podemos tener varios	
bloquea todos los hilos hilos por proceso No podemos calendari- No podemos tener varios	

Para revisión, preferentemente imprimir documentos compilados desde automultiple-choice.

procesos por hilo



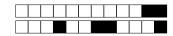
```
1
    char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
      int b=2;
10
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcap P \cap C$ manera segura en la lista circular.

manera segura en la nota encular.	\bigcirc 1	\bigcirc

43. Reescriba la función Queue::remove para que elimine de

manera segura en la lista circular.	$\bigcirc I \bigcirc P \bigcirc C$





Instrucciones. Marque las casillas (○) completamente sin salirse de ellas (por ejemplo ●). Responda a los siguientes cuestionamientos en las hojas que se le entregan marcando una única opción. Se utilizará factor de corrección 4 a 1

(las respuestas en blanco no se toman en cuenta). Las últim Parcial (2 pts.), e Incompleta (0 pts.). No llene las casillas de indicado.	
	← Marque su RUT sin código verificador (el número después del guión), y escriba sus nombres y apellidos abajo. Nombre(s) y apellido(s):
 1. ¿Cuál es el primer paso en la creación del proceso del programa prog? 	7. ¿Qué es la exclusión mutua? Objeto de sincronización que bloquea los hilos Propiedad de la concurrencia para mantener los hilos excluidos 7. ¿Qué es la exclusión mutua? Propiedad en la que solamente un hilo puede accesar simultáneamente Propiedad donde los hilos se bloquean mutuamente
 2. Modo de ejecución del procesador en el que verifica cada instrucción antes de ejecutarla. Modo dual	8. ¿Qué es una variable de condición? Variable de sincronización que permite la espera eficiente de un hilo Objeto de sincronización que implementa yield ción que permite la exclusión mutua Variable de sincronización que implementa yield ción que permite la exclusión mutua
 Stack Heap RAM Swap 4. ¿Qué característica del sistema operativo permite que opere 	 9. Es una señal síncrona al procesador que indica que ocurrió un evento que requiere su atención.
independientemente del hardware en la computadora? O Portabilidad O Ilusionista O Compartimiento de re- cursos 5. ¿Cuál de los siguientes elementos no es compartido por los hilos? O Stack O Código	10. ¿Qué es una operación atómica? Es una operación de bardware que no puede ser dividida Es una operación indivisible que no puede ser dividida Es una operación indivisible que no puede ser dividida Es una operación del kernel que no puede ser dividida
 Variables globales Gué es concurrencia? ○ Realizar múltiples actividades sin la interrupción de otras ○ Realizar múltiples actividades una después de la otra ○ Realizar múltiples actividades in la interrupción de otra 	11. ¿Cuál es el principal problema de los hilos a nivel de usuario? No podemos calendari- zar hilos dentro del pro- ceso El bloqueo del proceso, bloquea todos los hilos 12. ¿Cuál de los siguientes elementos no pertenece al TCB? Registros salvados Variables locales O No podemos tener varios hilos por proceso hilos por proceso Puntero al stack frame Estado del hilo



THE CENTER DE THE CENTER W			
13. Modo de ejecución del proces guna verificación. Modo inseguro Modo de usuario	Sador donde no se ejecuta nin- Modo kernel Modo dual	21. ¿Qué hace la función yield o Entrega voluntariamente el resto del ciclo en el procesador, y cambia el	de un hilo? Duerme al hilo por un tiempo determinado, y cambia el estado a espe-
14. ¿Cuándo se da una condición Varios hilos se ejecutan simultáneamente El estado de la ejecución de un programa depende del intercalado de diferentes hilos		estado a esperando (waiting) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished) 22. ¿Qué es la ejecución de una a	rando (waiting) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready)
15. ¿Cuál de las siguientes afirma spin wait?	ciones es verdadera respecto a	gidos?	
O Produce deadlocks O Es justificable en mono-	Es justificable en multi- procesadores	O Aplicación Proceso	○ Hilo○ Programa
procesadores $ {\bf 16.} \ \ {\bf ¿Qu\'e \ son \ las \ direcciones \ de \ m} $		23. Tipo de kernel en el que la fur de él, y el resto se encuentra a niv	
Memoria que reserva el sistema operativo para el uso de máquinas virtua- les	 Memoria adicional que crea el sistema operativo para engañar a las apli- caciones 	 Macrokernel 24. ¿Qué es el multi-hilado preve ○ Es cuando los hilos que 	Microkernel entivo?
 Capa de indirección que le da flexibilidad al siste- ma operativo para admi- nistrar la memoria 	Capa adicional que utiliza el sistema operativo en modo kernel	se ejecutan no pueden descalendarizarse Es cuando los hilos entregan el procesador vo-	Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos no
17. Es una señal asíncrona al pro-		luntariamente, no son in-	usan el procesador hasta
un evento que requiere su atención Señal de software	1. Interrupción	terrumpidos	que se les entrega indefi- nidamente
Señal de I/O	O Trampa	25. ¿Cuál es el estado de un hilo	en el que está listo para ejecu-
18. ¿Cuál de las siguientes no es criptor de archivo de UNIX? ○ Se cierran completamente después de usar	O Se abren antes de usar	tarse pero no está en el procesado Esperando (wait) Listo (ready)	
O Se escriben a través de bytes	O Se leen a través de un buffer	26. ¿Qué es un hilo? Una secuencia de ejecución que es calendarizada dependientemente	Varias secuencias de ejecución que pueden ser calendarizadas indepen-
 19. ¿Qué es un pipe? Es un buffer del kernel entre dos descriptores de archivos Es un buffer de usuario entre dos descriptores de 	Es un buffer de usuario entre dos procesosEs un buffer del kernel entre dos procesos	Varias secuencias de ejecución que pueden ser calendarizadas dependientemente	dientemente Una secuencia de ejecución que puede ser calendarizada independientemente
archivos 20. Tipo de kernel en el que la n éste se encuentra dentro de él.	nayoría de la funcionalidad de	27. ¿Cuál de las siguientes no es Comparte el código del programa	s una característica de un hilo? O Tiene un contador de programa propio
○ Monolítico○ Microkernel	○ Híbrido○ Macrokernel	O Tiene un segmento de datos propio	Tiene un stack propio
Wherokerner	Macrokerner	28. ¿Cuál de las siguientes es un Copiar el proceso padre completamente Copiar el proceso padre	a característica de fork? Copiar a la memoria el código del programa del proceso padre
		con privilegios distintos	Crear una copia del pro- ceso padre, pero no pue- de ser confiado igual que

él



29. ¿Qué es una llamada de siste Código del kernel que ejecuta código de usuario Funciones de hardware que llama el usuario	ema? Funciones del kernel que permiten al usuario accesar a recursos restringidos Llamada que hace el kernel para realizar una instrucción	<pre>36. ¿Cuántos procesos son cread guiente programa? main(int argc, forkthem(5); } void forkthem(ii if (n > 0){ fork();</pre>	char** argv){
30. Llamada del sistema de UN ciones el comunicarse entre sí paresumirlas.		forkthem(n-: } }	1);
fork	exec	\bigcirc 4	\bigcirc 30
○ wait	signal	\bigcirc 5	\bigcirc 16
Walt	Signal	Dodo al simulanto oódino	y
31. ¿Cuál de los siguientes event modo kernel a usuario?	tos no genera una transición de	Dado el siguiente código	
Ocontinuar después de	Cambiar a un proceso di-	main(int argc, int child == :	
una interrupción Creación de nuevo proceso	ferente Clamada I/O a un dispositivo	int x = 5; if (child == (x += 5;	0) {
32. ¿Qué hace la función wait d	o una variable de condición?	<pre>} else { child = forl</pre>	k ():
Atómicamente libera los	Mueve el hilo a la lista de	x += 10;	
seguros y mueve el hilo a	espera atómicamente	if (child)	{
la lista de espera	Libera el seguro global, y	x+=5;	
O Duerme el hilo atómica-	duerme al hilo		
mente mientras despier-		}	
ta al siguiente		37. ¿Cuántas copias distintas de	la variable x existen en la eje-
33. ¿Donde almacena el sistema o	operativo la toda la información	cución del programa anterior?	
sobre un proceso en particular?		\bigcirc 3	\bigcirc 4
() Heap	○ Stack	\bigcirc 2	\bigcirc 1
O Process control block	Memoria		
(PCB)	1 A A Y	38. ¿Cuál de los siguientes no es x al término de los procesos del pr	
34. En la creación del proceso de		\bigcirc 15	\bigcirc 20
el siguiente después de inicializar		\bigcirc 5	\bigcirc 10
Cargar prog en la memo-	O Inicializar el hardware	80 .0 /	9
ria O Copiar los argumentos	para que se ejecute el in- cio del programa	39. ¿Qué es un sistema operativo No es software, pero se	
en la memoria	Informar al calendariza-	encarga de administrar	Es un software que administra recursos
	dor que el programa se	recursos y usuarios	Es un software que admi-
	puede ejecutar	Es una capa de software	nistra usuarios
35. Región de la memoria reserva	de nor el sistema enerativo nora	que administra recursos	
mantener el estado de las variabl		y usuarios	
funciones.	os rocares durante la namada a	40. En la creación del proceso de	l programa prog toué paso es
RAM	Swap	el siguiente después de cargar pro	
Heap	Stack	O Inicializar el PCB en el	Copiar los argumentos
Treap	Diack	kernel	en la memoria
		O Informar al calendariza-	O Inicializar el hardware
		dor que el programa se puede ejecutar	para que se ejecute el in- cio del programa



```
1
    char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
      int b=2;
10
11
      foo();
12
```

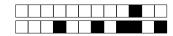
Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

tando la linea número 2. Utilice el número de línea para referirse a la dirección de cada instrucción.

OI OP OC

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

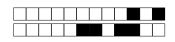
42. Reescriba la función Queue::insert para que inserte de manera segura en la lista circular.

	3	\sim	\sim	$\overline{}$
1				
ìì				
N				
- 1				
J				
A				

43. Reescriba la función Queue::remove para que elimine de manera segura en la lista circular.

manera segura en la lista circular.	\bigcirc I \bigcirc P \bigcirc C





Instrucciones. Marque las casillas (○) completamente sin salirse de ellas (por ejemplo ●). Responda a los siguientes

(las respuestas en blanco no se	e toman en cuenta). Las últim	o una única opción. Se utiliza las tres preguntas se evaluarán d la estas últimas tres preguntas, sir	de manera Completa (4 pts.),
$ \begin{array}{c cccc} & 1 & 1 & 1 & 1 \\ & 2 & 2 & 2 & 2 \\ & 3 & 3 & 3 & 3 \\ & 4 & 4 & 4 & 4 \\ & 5 & 5 & 5 & 5 \\ & 6 & 6 & 6 & 6 \\ & 7 & 7 & 7 & 7 \\ & 8 & 8 & 8 & 8 \end{array} $	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	← Marque su RUT sin o cador (el número después de escriba sus nombres y apell Nombre(s) y apellido(s):	lel guión), y
 ¿Qué es la exclusión mutua? Propiedad donde los hilos se bloquean mutuamente Propiedad de la concurrencia para mantener los hilos excluidos 	 Propiedad en la que solamente un hilo puede accesar simultáneamente Objeto de sincronización que bloquea los hilos 	 7. Tipo de kernel en el que la réste se encuentra dentro de él. Microkernel Monolítico 8. ¿Qué característica del sistem independientemente del hardware 	
 Tipo de kernel en el que la fur de él, y el resto se encuentra a ni Macrokernel Híbrido ¿Cuándo se da una condición Varios hilos se pueden intercalar en su ejecución simultánea Existe una sección crítica 	vel de usuario. Monolítico Microkernel	Portabilidad Ilusionista 9. ¿Qué es un sistema operativo No es software, pero se encarga de administrar recursos y usuarios Es una capa de software que administra recursos y usuarios	○ Virtualización○ Compartimiento de recursos
 4. Modo de ejecución del procesad verificación. Modo kernel Modo dual 5. ¿Qué es una variable de condi Variable de sincronización que permite la espera eficiente de un hilo 	Modo de usuarioModo inseguro	 10. ¿Cuál es el principal problema No podemos tener varios procesos por hilo No podemos calendarizar hilos dentro del proceso 11. ¿Qué es concurrencia? Realizar múltiples actividades simultáneamente Realizar múltiples actividades actividades actividades 	a de los hilos a nivel de usuario? No podemos tener varios hilos por proceso El bloqueo del proceso, bloquea todos los hilos Realizar una actividad sin la interrupción de otra
 Varialbe de sincronización que implementa yield 6. ¿Cuál es el estado de un hilo en pero no está en el procesador? ☐ Listo (ready) 	O Inicializado (init)	dades una después de la otra 12. Es una señal asíncrona al pro un evento que requiere su atenció Señal de I/O	Realizar múltiples actividades sin la interrupción de otras ocesador que indica que ocurrión. Señal de software
() Esperando (wait)	() Eiecutandose (running)	() Interrupción	() Trampa



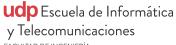
TACOLIA DE INCENIENA	
Jonde almacena el sistema operativo la toda la información sobre un proceso en particular? O Process control block (PCB) O Heap Dado el siguiente código main(int argc, char** argv){ int child == fork();	20. ¿Qué son las direcciones de memoria virtuales? Memoria que reserva el sistema operativo para el uso de máquinas virtuales Memoria adicional que crea el sistema operativo para engañar a las aplicaciones Capa adicional que en modo kernel Capa de indirección que le da flexibilidad al sistema operativo para administrar la memoria
<pre>int x = 5; if (child == 0) { x += 5; } else { child = fork(); x += 10; if (child) { x+=5; } } 14. ¿Cuál de los siguientes no es un valor que toma la variable x al término de los procesos del programa anterior?</pre>	21. ¿Cuál de los siguientes elementos no es compartido por los hilos? Stack Variables globales Código 22. ¿Qué es un pipe? Es un buffer del kernel entre dos descriptores de archivos Es un buffer de usuario entre dos descriptores de archivos Es un buffer de usuario entre dos procesos
5 0 15 20 15. ¿Cuántas copias distintas de la variable x existen en la ejecución del programa anterior? 3 0 1 2 4	23. En la creación del proceso del programa prog, ¿qué paso es el siguiente después de inicializar el espacio de memoria? O Inicializar el hardware para que se ejecute el incio del programa Informar al calendarizador que el programa se en la memoria puede ejecutar
16. ¿Cuál es el primer paso en la creación del proceso del programa prog? O Inicializar la memoria O Inicializar el hardware para que se ejecute el proceso desde el inicio 17. ¿Qué es una llamada de sistema? O Funciones de hardware que llama el usuario O Funciones del kernel que permiten al usuario acce- Crear e inicializar el PCB en el kernel Copiar prog a la memoria Llamada que hace el kernel para realizar una instrucción Código del kernel que	24. ¿Qué es un hilo? Una secuencia de ejecución que puede ser calendarizada independientemente Warias secuencias de ejecución que pueden ser calendarizadas independientemente 25. Modo de ejecución del procesador en el que verifica cada instrucción antes de ejecutarla. Modo de usuario Modo kernel
permiten al usuario accessar a recursos restringidos 18. Es una señal síncrona al procesador que indica que ocurrió un evento que requiere su atención. Señal de software Señal de I/O Interrupción	 ✓ Modo de usuario ✓ Modo kernel ✓ Modo seguro 26. ¿Cuál de las siguientes afirmaciones es verdadera respecto a spin wait? ✓ Es justificable en monoprocesadores ✓ No se debe de usar nunca ✓ Es justificable en multiprocesadores
19. ¿Cuál de los siguientes elementos no pertenece al <i>TCB</i> ? Caracteristic Estado del hilo Caracteristic Puntero al stack frame	27. ¿Qué hace la función wait de una variable de condición? Clibera el seguro global, y duerme al hilo Duerme el hilo atómicamente mientras despierta al siguiente seguros y mueve el hilo a Mueve el hilo a la lista de

la lista de espera

espera atómicamente



28. Llamada del sistema de UNI	X que le permite a las aplica-	35. ¿Qué hace la función yield	de un	hilo?
ciones el comunicarse entre sí para resumirlas.	a terminarlas, suspenderlas, o	O Entrega voluntariamente el resto del ciclo en el	\circ	Duerme al hilo por un tiempo determinado, y
exec	○ signal	procesador, y cambia el estado a terminado (fi-		cambia el estado a esperando (waiting)
wait29. ¿Cuántos procesos son creado	os durante la ejecución del si-	nished)	\bigcirc	Entrega voluntariamente el resto del ciclo en el
guiente programa?	s durante la ejectición del si-	 Entrega voluntariamente el resto del ciclo en el procesador, y cambia el 		procesador, y cambia el estado a esperando (wai-
<pre>main(int argc, c forkthem(5);</pre>	har** argv){	estado a listo $(ready)$		ting)
}		36. ¿Qué es una operación atóm	ica?	
<pre>void forkthem(in if (n > 0){</pre>	t n) {	Es una operación del ker-	O	Es una operación indivi-
fork();		nel que no puede ser di- vidida		sible que no puede ser di- vidida
forkthem(n-1);	Es una operación que		Es una operación de
}		deshabilita las interrup-		hardware que no puede
	\bigcap 4	ciones		ser dividida
$ \bigcirc 30 \\ \bigcirc 5 $	\bigcirc 16	37. ¿Cuál de las siguientes es un	ia cara	acterística de fork?
\bigcirc 5	O 10	O Copiar el proceso padre		Copiar a la memoria el
30. En la creación del proceso del el siguiente después de cargar prog		completamente		código del programa del proceso padre
Inicializar el hardware	Inicializar el PCB en el	Opiar el proceso padre con privilegios distintos	\bigcirc	Crear una copia del pro-
para que se ejecute el in-	kernel	con privilegios distilitos	\circ	ceso padre, pero no pue-
cio del programa	Ocopiar los argumentos			de ser confiado igual que
O Informar al calendariza-	en la memoria			él
dor que el programa se		38. ¿Cuál de las siguientes no e	s una	característica de un hilo?
puede ejecutar		Comparte el código del	\bigcirc	Tiene un contador de
31. ¿Cuál de las siguientes no es	una característica de un des-	programa		programa propio
criptor de archivo de UNIX?	- A A	O Tiene un segmento de	\bigcirc	Tiene un stack propio
O Se leen a través de un	Se cierran completamen-	datos propio		
buffer	te después de usar	39. ¿Qué es el multi-hilado prev	entivo	?
Se escriben a través de bytes	O Se abren antes de usar	O Es cuando los hilos en-	\bigcirc	Es cuando los hilos no
		tregan el procesador vo-		usan el procesador hasta
32. Región de la memoria reserv para alojar estructuras de datos qu		luntariamente, no son in- terrumpidos		que se les entrega indefi- nidamente
○ Swap	Heap	Es cuando los hilos que	\circ	Es cuando los hilos que
RAM	Stack	se ejecutan no pueden		se ejecutan pueden ser
		descalendarizarse		cambiados indistinta- mente
33. ¿Qué es la ejecución de una apgidos?	olicación con permisos restrin-	40 0 0 1 1 1 1 1 1 1		
	O B	40. ¿Cuál de los siguientes event modo kernel a usuario?	os no	genera una transición de
○ Hilo	O Programa	Creación de nuevo proce-	\bigcirc	Cambiar a un proceso di-
(Aplicación	() Proceso	so	\cup	ferente
34. Región de la memoria reservado mantener el estado de las variables funciones.		Ocontinuar después de una interrupción	\circ	Llamada I/O a un dispositivo
Heap	○ Swap			
○ RAM	O Stack			
<u> </u>	<u></u>			



FACULTAD DE INGENIERÍA

41. Dado el siguiente código del programa main

```
1
   char bar(){
2
     int b = 2;
3
      return 'a';
4 }
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
10
11
      foo();
12
```

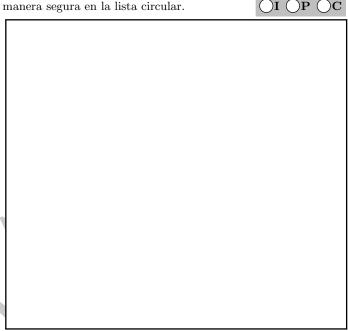
Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcirc I \bigcirc P \bigcirc C$

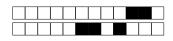


El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

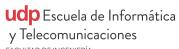
43.	Reescriba la función Qu	eue::remove	para qu	e elim	ine de
man	era segura en la lista circ	ular.	\bigcirc I	$\bigcirc \mathbf{P}$	$\bigcirc \mathbf{C}$





Instrucciones. Marque las casillas (○) completamente sin salirse de ellas (por ejemplo ●). Responda a los siguientes

(las respuestas en blanco no se	e toman en cuenta). Las últim	o una única opción. Se utiliza las tres preguntas se evaluarán a le estas últimas tres preguntas, sir	le manera Completa (4 pts.),
$ \begin{array}{c c} \hline \bigcirc 1 \bigcirc 1 \bigcirc 1 \bigcirc 1 \\ \hline \bigcirc 2 \bigcirc 2 \bigcirc 2 \bigcirc 2 \end{array} $	0 0 0 0 0 0 $0 1 1 1 1$ $0 0 0 0$ $0 0 0$	← Marque su RUT sin c cador (el número después d escriba sus nombres y apell	lel guión), y
$igcirc$ 4\\ 4\\ 4\\ 05\\ 5\\ 5\\ 5\\ 5\\ 6\\ 6\\ 6\\ 6\\ 6\\	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	Nombre(s) y apellido(s):	
1. Tipo de kernel en el que la fur de él, y el resto se encuentra a ni		7. ¿Cuál de las siguientes no escriptor de archivo de UNIX?	s una característica de un des-
Híbrido Monolítico	Macrokernel Microkernel	Se escriben a través de bytes Se leen a través de un	Se abren antes de usarSe cierran completamente después de usar
2. ¿Cuál de las siguientes es una		buffer	-
 Copiar el proceso padre completamente Crear una copia del proceso padre, pero no puede ser confiado igual que él 	 Copiar a la memoria el código del programa del proceso padre Copiar el proceso padre con privilegios distintos 	8. ¿Qué hace la función yield de O Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting)	e un hilo? Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (wai- ting)
3. Región de la memoria reservado mantener el estado de las variable funciones.RAMStack		Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready)	Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished)
4. Región de la memoria reservac		9. ¿Qué es un hilo?	O Una gaguancia da ciagu
alojar estructuras de datos que e	l proceso pueda necesitar. Heap Swap	Varias secuencias de ejecución que pueden ser calendarizadas dependientemente	Una secuencia de ejecución que puede ser calendarizada independientemente
5. ¿Cuál es el principal problema El bloqueo del proceso, bloquea todos los hilos	O No podemos calendarizar hilos dentro del pro-	 Una secuencia de ejecu- ción que es calendarizada dependientemente 	 Varias secuencias de eje- cución que pueden ser calendarizadas indepen- dientemente
No podemos tener varios procesos por hilo	ceso No podemos tener varios hilos por proceso	10. ¿Qué característica del sistem independientemente del hardware	
6. Tipo de kernel en el que la réste se encuentra dentro de él.	* *	○ Virtualización○ Portabilidad	 Ilusionista Compartimiento de recursos
	○ Macrokernel○ Híbrido	11. ¿Cuál es el primer paso en l	a creación del proceso del pro-
-	-	grama prog? O Inicializar el hardware para que se ejecute el proceso desde el inicio	Inicializar la memoriaCopiar prog a la memoria
		Crear e inicializar el PCB en el kernel	

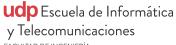


FACUITAD	DF	INGENIERÍA

12. ¿Qué hace la función wait de Clibera el seguro global, y	O Mueve el hilo a la lista de	20. ¿Cuándo se da una condición El estado de la ejecución	O Varios hilos se pueden in-
duerme al hilo Duerme el hilo atómicamente mientras despierta al siguiente	espera atómicamente Atómicamente libera los seguros y mueve el hilo a la lista de espera	de un programa depende del intercalado de diferentes hilos Varios hilos se ejecutan simultáneamente	tercalar en su ejecución simultánea Existe una sección críti- ca
13. ¿Cuál de los siguientes eleme Estado del hilo	ntos no pertenece al TCB ? Registros salvados	21. Modo de ejecución del proces guna verificación.	ador donde no se ejecuta nin-
O Variables locales	O Puntero al stack frame	Modo kernel	Modo inseguro
14. ¿Qué es una variable de cond	lición?	Modo de usuario	O Modo dual
Objeto de sincronización que implementa yield en modo usuario	O Varialbe de sincronización que implementa yield	22. Es una señal asíncrona al produn evento que requiere su atención	1.
O Variable de sincroniza- ción que permite la ex-	O Variable de sincronización que permite la espe-	○ Señal de I/O○ Trampa	✓ Interrupción✓ Señal de software
clusión mutua	ra eficiente de un hilo		
15. ¿Qué es concurrencia?		23. Es una señal síncrona al proc un evento que requiere su atención	
Realizar múltiples actividades simultáneamente	Realizar una actividad sin la interrupción de	O Interrupción	○ Trampa
Realizar múltiples activi-	otra	O Señal de software	O Señal de I/O
dades sin la interrupción de otras	Realizar múltiples actividades una después de la otra	24. ¿Donde almacena el sistema op sobre un proceso en particular?	perativo la toda la información
16. En la creación del proceso de		(PCB) Process control block	O Stack
el siguiente después de inicializar		(1 CB) Memoria	(Heap
Cargar prog en la memo-	O Informar al calendariza-		
ria Copiar los argumentos	dor que el programa se puede ejecutar	25. ¿Cuál de las siguientes afirma spin wait?	ciones es verdadera respecto a
en la memoria	O Inicializar el hardware	Es justificable en mono-	O No se debe de usar nunca
	para que se ejecute el in- cio del programa	procesadores Es justificable en multi-	O Produce deadlocks
17. Modo de ejecución del proce	esador en el que verifica cada	procesadores	
instrucción antes de ejecutarla.		26. ¿Cuál de las siguientes no es	
Modo seguro	✓ Modo dual✓ Modo kernel	Tiene un contador de programa propio	Tiene un segmento de datos propio
Modo de usuario	J	Comparte el código del	Tiene un stack propio
18. ¿Cuántos procesos son cread guiente programa?	os durante la ejecución del si-	programa	
guiente programa:		27. ¿Cuál de los siguientes evento	s no genera una transición de
main(int argc, o	char** argv){	modo kernel a usuario? Cambiar a un proceso di-	Creación de nuevo proce-
<pre>forkthem(5); }</pre>		ferente	so
<pre>void forkthem(ir if (n > 0){</pre>	nt n){	Continuar después de una interrupción	O Llamada I/O a un dispositivo
fork();	()	_	
forkthem(n-1)	1);	28. En la creación del proceso del el siguiente después de cargar prog	
}		Informar al calendariza-	Inicializar el hardware
\bigcirc 5	\bigcirc 30	dor que el programa se	para que se ejecute el in-
O 16	\bigcirc 4	puede ejecutar Copiar los argumentos	cio del programa () Inicializar el PCB en el
19. ¿Cuál es el estado de un hilo tarse pero no está en el procesado		Copiar los argumentos en la memoria	kernel
O Inicializado (init)	\bigcirc Esperando (wait)		
\bigcirc Listo $(ready)$	\bigcirc Ejecutandose (running)		

ud	O Escuela de Informática
,	lecomunicaciones

29. ¿Qué es la exclusión mutua? O Propiedad de la concurrencia para mantener los hilos excluidos O Propiedad donde los hilos se bloquean mutuamente 30. ¿Qué es un pipe? O Es un buffer del kernel	Propiedad en la que solamente un hilo puede accesar simultáneamente Objeto de sincronización que bloquea los hilos Es un buffer del kernel	<pre>main(int argc, cl int child == for int x = 5; if (child == 0) x += 5; } else { child = fork x += 10; if (child) { x+=5; }</pre>	ork();) {
entre dos descriptores de archivos Es un buffer de usuario entre dos procesos	entre dos procesos Es un buffer de usuario entre dos descriptores de archivos	} 36. ¿Cuál de los siguientes no es x al término de los procesos del pro	
31. Llamada del sistema de UNIX que ciones el comunicarse entre sí para ter resumirlas.		○ 15 ○ 10	0 5 0 20
() wait () signal	exec fork	37. ¿Cuántas copias distintas de la cución del programa anterior?	a variable \mathbf{x} existen en la eje-
32. ¿Qué es el multi-hilado preventivo Es cuando los hilos que se ejecutan no pueden descalendarizarse Es cuando los hilos entregan el procesador voluntariamente, no son interrumpidos	es es cuando los hilos no usan el procesador hasta que se les entrega indefinidamente Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente	2 1 38. ¿Qué es la ejecución de una ap gidos? Programa Hilo 39. ¿Cuál de los siguientes elementalos?	AplicaciónProceso
 33. ¿Qué es una operación atómica? Es una operación indivisible que no puede ser dividida Es una operación del kernel que no puede ser dividida 	Es una operación que deshabilita las interrup- ciones Es una operación de hardware que no puede ser dividida	Stack Código 40. ¿Qué son las direcciones de moderne de la sistema operativo para engañar a las apli-	 Variables globales Heap emoria virtuales? Capa de indirección que le da flexibilidad al sistema operativo para admi-
34. ¿Qué es un sistema operativo? ○ Es una capa de software que administra recursos y usuarios ○ Es un software que administra recursos	Es un software que administra usuarios No es software, pero se encarga de administrar recursos y usuarios	caciones Capa adicional que utiliza el sistema operativo en modo kernel	nistrar la memoria Memoria que reserva el sistema operativo para el uso de máquinas virtuales
 35. ¿Qué es una llamada de sistema? Funciones de hardware que llama el usuario Llamada que hace el kernel para realizar una instrucción Dado el siguiente código 	Código del kernel que ejecuta código de usuario Funciones del kernel que permiten al usuario acce- sar a recursos restringi- dos		



FACULTAD DE INGENIERÍA

41. Dado el siguiente código del programa main

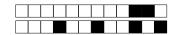
```
1
    char bar(){
2
     int b = 2;
3
     return 'a';
4 }
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){}
     int b=2;
10
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

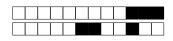
42. Reescriba la función Queue::insert para que inserte de \bigcirc I \bigcirc P \bigcirc C manera segura en la lista circular.

	-	
r		
1		i
1		i
1		i
1		i
1		i
ı		
ı		
п		
1		i
ı		
ı		
ı		
ı		
ı		
ı		
ı		
1		i
1		i
1		i
1		i
1		
1		
1		
ı		
ı		
ı		
ı		
ı		
1		
1		i
ı		
ı		
ı		
ı		
ı		
ı		
1		
1		
1		
ì		
ì		
1		
1		
ы		
u		
М		
-1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
1		
ı		
1		
1		
1		
1		
J		
ı		
ı		
٦		
1		

43. Reescriba la función Queue::remove para que elimine de 0-0-0-

nanera segura en la lista circular.	$\bigcirc I \bigcirc P \bigcirc C$





Instrucciones. Marque las casillas (○) completamente sin salirse de ellas (por ejemplo ●). Responda a los siguientes cuestionamientos en las hojas que se le entregan marcando una única opción. Se utilizará factor de corrección 4 a 1 (las respuestas en blanco no se toman en cuenta). Las últimas tres preguntas se evaluarán de manera Completa (4 pts.), Parcial (2 pts.), e Incompleta (0 pts.). No llene las casillas de estas últimas tres preguntas, sino que responda en el espacio indicado.

$\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0$
$\bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1$
$\bigcirc 2 \bigcirc 2$
$\bigcirc 3 \bigcirc 3$
$\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4$
$\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5$
$\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6$
$\bigcirc 7 \bigcirc 7$
$\bigcirc 8 \bigcirc 8$
$\bigcirc 9 \bigcirc 9$

Dado el siguiente código

```
main(int argc, char** argv){
  int child == fork();
  int x = 5;
  if (child == 0) {
    x += 5;
  } else {
    child = fork();
    x += 10;
    if (child) {
        x+=5;
    }
  }
}
```

1. ¿Cuántas copias distintas de la variable x existen en la ejecución del programa anterior?

 $\bigcirc \ 2 \\ \bigcirc \ 4$ $\bigcirc \ 1$

2. ¿Cuál de los siguientes ${\bf no}$ es un valor que toma la variable ${\bf x}$ al término de los procesos del programa anterior?

○ 10○ 5○ 2

3. Región de la memoria reservada por el sistema operativo para mantener el estado de las variables locales durante la llamada a funciones.

◯ Heap
 ◯ Stack
 ◯ RAM
 ◯ Swap

4. ¿Qué es una variable de condición?

Varialbe de sincronización que implementa yield

O Variable de sincronización que permite la espera eficiente de un hilo O Variable de sincronización que permite la exclusión mutua

Objeto de sincronización que implementa yield en modo usuario ← Marque su RUT sin código verificador (el número después del guión), y escriba sus nombres y apellidos abajo.

Nombre(s) y apellido(s):

5. ¿Qué es una llamada de sistema?

- Llamada que hace el kernel para realizar una instrucción
 - Funciones de hardware que llama el usuario
- Funciones del kernel que permiten al usuario accesar a recursos restringidos
- O Código del kernel que ejecuta código de usuario

6. ¿Qué hace la función yield de un hilo?

- Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (waiting)
- Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready)
- O Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting)
- Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished)

7. ¿Cuántos procesos son creados durante la ejecución del siguiente programa?

```
main(int argc, char** argv){
    forkthem(5);
}

void forkthem(int n){
    if (n > 0){
        fork();
        forkthem(n-1);
    }
}

4

16
```

8. ¿Cuál de las siguientes afirmaciones es verdadera respecto a $spin\ wait?$

Produce deadlocks	O No se debe de usar nunca
Es justificable en multi-	O Es justificable en mono-
procesadores	procesadores



16. ¿Qué son las direcciones de memoria virtuales?

() Capa adicional que uti-

Memoria adicional que

en modo kernel

caciones

liza el sistema operativo

crea el sistema operativo

para engañar a las apli-

Liscuela de Illion			
y Telecomunicaciones			
FACULTAD DE INGENIERÍA			
9. Modo de ejecución del procesa instrucción antes de ejecutarla.	dor en el que verifica cada	17. ¿Cuál es el principal problema de los hilos a nivel de usuario No podemos calendari- El bloqueo del proceso	
O Modo dual (Modo de usuario	zar hilos dentro del pro-	
○ Modo seguro (Modo kernel	ceso No podemos tener varios No podemos tener varios hilos por proceso	S
10. ¿Cuál es el estado de un hilo en tarse pero no está en el procesador?	el que está listo para ejecu-	procesos por hilo 18. ¿Qué hace la función wait de una variable de condición?	
\bigcirc Listo $(ready)$ (Inicializado (init)	Mueve el hilo a la lista de Duerme el hilo atómica	
Esperando (wait)	Ejecutandose (running)	espera atómicamente mente mientras despier	
11. ¿Cuál de las siguientes no es un	na característica de un hilo?	Atómicamente libera los ta al siguiente	
Tiene un contador de (programa propio	Comparte el código del programa	seguros y mueve el hilo a () Libera el seguro global, la lista de espera () duerme al hilo	У
Tiene un stack propio (Tiene un segmento de	19. ¿Qué es un hilo?	
	datos propio	Varias secuencias de Varias secuencias de eje ejecución que pue- cución que pueden se	
12. ¿Cuál es el primer paso en la c grama prog?	reación del proceso del pro-	den ser calendarizadas calendarizadas indeper dependientemente dientemente	
O Inicializar la memoria	Crear e inicializar el	O Una secuencia de ejecu-	
Ocopiar prog a la memo-	PCB en el kernel	ción que puede ser calen-	a
ria	Inicializar el hardware para que se ejecute el	darizada independiente- dependientemente	
	proceso desde el inicio	mente	
10 0 / 11 11 11 11 11	1	20. ¿Qué es una operación atómica?	
13. ¿Qué es un sistema operativo?		() Es una operación del ker- () Es una operación que des la	
O No es software, pero se encarga de administrar	Es un software que administra recursos	nel que no puede ser di- vidida deshabilita las interrup ciones)-
recursos y usuarios (Es un software que admi-	Es una operación de Es una operación indiv	i-
Es una capa de software	nistra usuarios	hardware que no puede sible que no puede ser d	
que administra recursos		ser dividida vidida	
y usuarios		21. En la creación del proceso del programa prog, ¿qué paso e	es
14. ¿Qué es el multi-hilado prevent	ivo?	el siguiente después de inicializar el espacio de memoria?	
Es cuando los hilos que (Es cuando los hilos en-	Cargar prog en la memo- Copiar los argumento	S
se ejecutan pueden ser	tregan el procesador vo-	ria en la memoria O Informar al calendariza- Inicializar el hardwar	
cambiados indistinta-	luntariamente, no son in-	O Informar al calendariza- dor que el programa se Dinicializar el hardwar para que se ejecute el ir	
mente	terrumpidos	puede ejecutar cio del programa	
Es cuando los hilos que se ejecutan no pueden	Es cuando los hilos no		
descalendarizarse	usan el procesador hasta que se les entrega indefi-	22. ¿Cuál de las siguientes es una característica de fork? Copiar a la memoria el Crear una copia del pro	
	nidamente	Copiar a la memoria el Crear una copia del pro código del programa del ceso padre, pero no pue	
15. ¿Qué es un pipe?		proceso padre de ser confiado igual qu	
Es un buffer del kernel (Es un buffer de usuario	Copiar el proceso padre él	
entre dos descriptores de	entre dos descriptores de	completamente	
archivos	archivos	con privilegios distintos	í
Es un buffer de usuario (Es un buffer del kernel	23. Modo de ejecución del procesador donde no se ejecuta nir	1-
entre dos procesos	entre dos procesos	guna verificación.	

() Capa de indirección que

nistrar la memoria

les

le da flexibilidad al siste-

ma operativo para admi-

Memoria que reserva el

sistema operativo para el

uso de máquinas virtua-

Modo dual

O Proceso

Aplicación

gidos?

Modo inseguro

O Modo kernel

O Hilo

O Programa

24. ¿Qué es la ejecución de una aplicación con permisos restrin-

O Modo de usuario



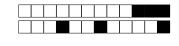
25. En la creación del proceso del el siguiente después de cargar prog O Inicializar el hardware			35. Llamada del sistema de UNI ciones el comunicarse entre sí para resumirlas.	
para que se ejecute el in-	\sim	kernel	exec	○ wait
cio del programa	\bigcirc	Informar al calendariza-	of fork	Signal
Copiar los argumentos en la memoria		dor que el programa se puede ejecutar	36. ¿Cuál de los siguientes elementilos?	ntos no es compartido por los
26. ¿Qué es concurrencia? Realizar múltiples activi-	\bigcirc	Realizar una actividad	O Stack	O Variables globales
dades simultáneamente	\cup	sin la interrupción de	Heap	Código
Realizar múltiples activi-		otra		Q
dades una después de la otra	\bigcirc	Realizar múltiples actividades sin la interrupción	37. Es una señal asíncrona al produn evento que requiere su atención	
		de otras	○ Señal de I/O	Trampa
27. ¿Qué es la exclusión mutua?			O Señal de software	○ Interrupción
Objeto de sincronización que bloquea los hilos	0	Propiedad de la concu- rrencia para mantener	38. Tipo de kernel en el que la méste se encuentra dentro de él.	nayoría de la funcionalidad de
O Propiedad en la que sola-	\bigcirc	los hilos excluidos Propiedad donde los hi-	() Monolítico	Microkernel
mente un hilo puede ac- cesar simultáneamente	\cup	los se bloquean mutua-	Macrokernel	Híbrido
cesar simurtaneamente		mente	Wiacrokerner	Tribrido
28. ¿Qué característica del sistema independientemente del hardware e	_		39. ¿Cuál de los siguientes evento modo kernel a usuario?	
O Portabilidad	\bigcirc	Virtualización	Cambiar a un proceso di- ferente	Creación de nuevo proceso
Compartimiento de re-	$\tilde{\bigcirc}$	Ilusionista	Clamada I/O a un dispo-	Continuar después de
cursos			sitivo	una interrupción
29. Tipo de kernel en el que la fund de él, y el resto se encuentra a nive			40. ¿Donde almacena el sistema op sobre un proceso en particular?	perativo la toda la información
○ Híbrido	\bigcirc	Microkernel	Heap	O Process control block
○ Macrokernel	\bigcirc	Monolítico	() Memoria	(PCB)
30. Es una señal síncrona al proc un evento que requiere su atención		or que indica que ocurrió	,	() Stack
O Interrupción				
/	()	Señal de software		
() Trampa	\bigcirc	Señal de software Señal de I/O		
TrampaRegión de la memoria reserv		Señal de I/O por el sistema operativo		
31. Región de la memoria reserv para alojar estructuras de datos qu		Señal de I/O por el sistema operativo proceso pueda necesitar.		
31. Región de la memoria reserv para alojar estructuras de datos qu O Swap		Señal de I/O por el sistema operativo proceso pueda necesitar. RAM		
31. Región de la memoria reserv para alojar estructuras de datos qu		Señal de I/O por el sistema operativo proceso pueda necesitar.		
31. Región de la memoria reserv para alojar estructuras de datos qu O Swap	ue el	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap		
31. Región de la memoria reserv para alojar estructuras de datos qu	ue el	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamen-		
31. Región de la memoria reserv para alojar estructuras de datos qu Swap Stack 32. ¿Cuál de las siguientes no es criptor de archivo de UNIX? Se escriben a través de bytes	ue el	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar		
31. Región de la memoria reserv para alojar estructuras de datos qu Swap Stack 32. ¿Cuál de las siguientes no es criptor de archivo de UNIX? Se escriben a través de	ue el	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un		
31. Región de la memoria reserv para alojar estructuras de datos qu Swap Stack 32. ¿Cuál de las siguientes no es criptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar	una	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer		
 31. Región de la memoria reserve para alojar estructuras de datos que Swap Stack 32. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición 	una	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera?		
31. Región de la memoria reserve para alojar estructuras de datos qua Swap Stack 32. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición El estado de la ejecución	una	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera? Varios hilos se ejecutan		
 31. Región de la memoria reserve para alojar estructuras de datos que Swap Stack 32. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición 	una	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera?		
 31. Región de la memoria reserve para alojar estructuras de datos que Swap Stack 32. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición El estado de la ejecución de un programa depende del intercalado de diferentes hilos 	una	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera? Varios hilos se ejecutan simultáneamente		
31. Región de la memoria reserve para alojar estructuras de datos que Swap Stack 32. ¿Cuál de las siguientes no es criptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición El estado de la ejecución de un programa depende del intercalado de diferentes hilos Existe una sección críti-	una	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera? Varios hilos se ejecutan simultáneamente Varios hilos se pueden in-		
31. Región de la memoria reserve para alojar estructuras de datos quanto Swap Stack 32. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición El estado de la ejecución de un programa depende del intercalado de diferentes hilos Existe una sección crítica	una de c	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera? Varios hilos se ejecutan simultáneamente Varios hilos se pueden intercalar en su ejecución simultánea		
31. Región de la memoria reserve para alojar estructuras de datos que Swap Stack 32. ¿Cuál de las siguientes no es criptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición El estado de la ejecución de un programa depende del intercalado de diferentes hilos Existe una sección críti-	una de c	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera? Varios hilos se ejecutan simultáneamente Varios hilos se pueden intercalar en su ejecución simultánea		
31. Región de la memoria reserve para alojar estructuras de datos quanto Swap Stack 32. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 33. ¿Cuándo se da una condición El estado de la ejecución de un programa depende del intercalado de diferentes hilos Existe una sección crítica	una de c	Señal de I/O por el sistema operativo proceso pueda necesitar. RAM Heap característica de un des- Se cierran completamente después de usar Se leen a través de un buffer arrera? Varios hilos se ejecutan simultáneamente Varios hilos se pueden intercalar en su ejecución simultánea		

Para revisión, preferentemente imprimir documentos compilados desde automultiple-choice.



```
1
    char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
      int b=2;
10
11
      foo();
12
```

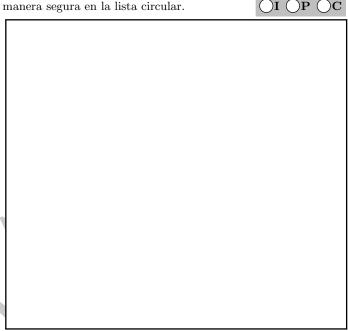
Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse a la dirección de cada instrucción.



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de manera segura en la lista circular.



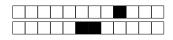
El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

manera segura en la lista circular.	\bigcirc I \bigcirc P \bigcirc C

43. Reescriba la función Queue::remove para que elimine de





cuestionamientos en las hojas qua (las respuestas en blanco no se	ue se le entregan marcando toman en cuenta). <i>Las últim</i>	salirse de ellas (por ejemplo •) una única opción. Se utilizara as tres preguntas se evaluarán de estas últimas tres preguntas, sino	á factor de corrección 4 a 1 e manera Completa (4 pts.),
$\begin{array}{c cccc} \bigcirc 0 \bigcirc $	1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5 6 6 6 6 6 7 7 7 7 7 8 8 8 8 8	← Marque su RUT sin cocador (el número después de escriba sus nombres y apellido Nombre(s) y apellido(s):	el guión), y
1. ¿Qué es la ejecución de una apgidos?	licación con permisos restrin-	7. Es una señal síncrona al proce un evento que requiere su atención	
AplicaciónPrograma	O Proceso Hilo	○ Señal de I/O○ Trampa	Señal de softwareInterrupción
 ¿Qué es una llamada de sistem Código del kernel que ejecuta código de usuario Llamada que hace el kernel para realizar una instrucción ¿Cuál de los siguientes eventos modo kernel a usuario? Cambiar a un proceso diferente Continuar después de una interrupción Modo de ejecución del proces instrucción antes de ejecutarla. Modo de usuario Modo dual ¿Cuándo se da una condición de Existe una sección crítica El estado de la ejecución 	Funciones de hardware que llama el usuario Funciones del kernel que permiten al usuario accesar a recursos restringidos no genera una transición de Llamada I/O a un dispositivo Creación de nuevo proceso sador en el que verifica cada Modo kernel Modo seguro	8. En la creación del proceso del el siguiente después de cargar progonico de la lista de PCB en el kernel Informar al calendarizador que el programa se puede ejecutar 9. ¿Qué es la exclusión mutua? Objeto de sincronización que bloquea los hilos Propiedad donde los hilos se bloquean mutuamente 10. ¿Qué hace la función wait de Duerme el hilo atómicamente mientras despierta al siguiente Atómicamente libera los seguros y mueve el hilo a la lista de espera 11. ¿Qué es concurrencia?	programa prog, ¿qué paso es g en la memoria? Inicializar el hardware para que se ejecute el incio del programa Copiar los argumentos en la memoria Propiedad en la que solamente un hilo puede accesar simultáneamente Propiedad de la concurrencia para mantener los hilos excluidos
de un programa depende del intercalado de diferentes hilos 6. ¿Qué característica del sistema independientemente del hardware del Compartimiento de recursos	en la computadora? () Ilusionista	 Realizar múltiples actividades sin la interrupción de otras Realizar múltiples actividades una después de la otra 	Realizar múltiples actividades simultáneamente Realizar una actividad sin la interrupción de otra
O Portabilidad	(Virtualización	12. ¿Donde almacena el sistema op sobre un proceso en particular? Stack Heap	OPROCESS control block (PCB) Memoria



bloquea todos los hilos

FACULTAD DE INGENIERÍA	
13. ¿Cuál de las siguientes es una característica de fork? Copiar a la memoria el Copiar el proceso padre	20. ¿Cuántos procesos son creados durante la ejecución del siguiente programa?
código del programa del proceso padre Crear una copia del proceso padre, pero no puede ser confiado igual que él completamente completamente con privilegios distintos	<pre>main(int argc, char** argv){ forkthem(5); } void forkthem(int n){ if (n > 0){ fork(); }</pre>
14. Es una señal asíncrona al procesador que indica que ocurrió un evento que requiere su atención.	forkthem(n-1); }
Señal de I/O Trampa	}
○ Interrupción ○ Señal de software	$ \begin{array}{ccc} $
Dado el siguiente código	21. ¿Cuál de los siguientes elementos no es compartido por los
<pre>main(int argc, char** argv){ int child == fork(); int x = 5; if (child == 0) { x += 5; } else { child = fork(); x += 10; if (child) { x+=5;</pre>	hilos? Variables globales Stack Código 22. ¿Cuál de las siguientes no es una característica de un hilo? Tiene un contador de Comparte el código del programa propio programa Tiene un stack propio Tiene un segmento de
} }	datos propio
} 15. ¿Cuántas copias distintas de la variable x existen en la ejecución del programa anterior? 1 2 3 16. ¿Cuál de los siguientes no es un valor que toma la variable x al término de los procesos del programa anterior?	23. ¿Qué es el multi-hilado preventivo? Es cuando los hilos que se ejecutan no pueden descalendarizarse Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos que terrumpidos Es cuando los hilos no usan el procesador hasta que se les entrega indefinidamente
$ \begin{array}{ccc} \bigcirc 20 & \bigcirc 5 \\ \bigcirc 15 & \bigcirc 10 \end{array} $	24. ¿Qué es un hilo? Varias secuencias de O Varias secuencias de ejecicución que puedon ser
17. Tipo de kernel en el que la mayoría de la funcionalidad de éste se encuentra dentro de él. Monolítico Macrokernel 18. ¿Qué hace la función yield de un hilo?	den ser calendarizadas dependientemente Una secuencia de ejecución que es calendarizada dependientemente calendarizadas independientemente Una secuencia de ejecución que puede ser calendarizada independientemente
Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready)	25. ¿Qué es un sistema operativo? Es una capa de software que administra recursos encarga de administrar y usuarios Es un software que administra usuarios Es un software que administra recursos Es un software que administra recursos 26. Región de la memoria reservada por el sistema operativo para mantener el estado de las variables locales durante la llamada a funciones.
 19. ¿Cuál es el principal problema de los hilos a nivel de usuario? No podemos tener varios procesos por hilo El bloqueo del proceso, No podemos tener varios hilos por proceso No podemos calendari- 	○ Stack○ Swap○ RAM

zar hilos dentro del pro-

ceso



27. ¿Cuál de las siguientes afirmac spin wait?	ciones es verdadera respecto a	36. En la creación del proceso de el siguiente después de inicializar	
Es justificable en multiprocesadores Produce deadlocks	 Es justificable en monoprocesadores No se debe de usar nunca	O Informar al calendariza- dor que el programa se puede ejecutar	Copiar los argumentos en la memoriaInicializar el hardware
28. ¿Cuál de los siguientes elemen	ntos no pertenece al <i>TCB</i> ?	Cargar prog en la memo- ria	para que se ejecute el in- cio del programa
O Puntero al stack frame Variables locales	Estado del hiloRegistros salvados	37. Llamada del sistema de UN ciones el comunicarse entre sí pa resumirlas.	
29. Modo de ejecución del proces guna verificación.	ador donde no se ejecuta nin-	Signal	exec
Modo de usuario	O Modo inseguro	O fork	Owait
○ Modo dual	Modo kernel	38. ¿Cuál es el estado de un hilo tarse pero no está en el procesado	
30. ¿Qué es una operación atómic		Esperando (wait)	O Inicializado (init)
Es una operación de hardware que no puede	Es una operación del ker- nel que no puede ser di-	Ejecutandose (running)	Listo (ready)
ser dividida Es una operación indivi-	vidida O Es una operación que	39. Tipo de kernel en el que la fu de él, y el resto se encuentra a ni	
sible que no puede ser di- vidida	deshabilita las interrup- ciones	O Macrokernel	○ Híbrido
31. Región de la memoria reserv	vada por el sistema operativo	O Monolítico	O Microkernel
para alojar estructuras de datos qu	-	40. ¿Cuál de las siguientes no e	es una característica de un des-
O Heap	O Swap	criptor de archivo de UNIX?	
○ Stack	○ RAM	O Se abren antes de usar	O Se escriben a través de
32. ¿Cuál es el primer paso en la grama prog?	creación del proceso del pro-	O Se leen a través de un buffer	bytes Se cierran completamen-
Crear e inicializar el PCB en el kernel	O Inicializar el hardware para que se ejecute el		te después de usar
O Inicializar la memoria	proceso desde el inicio		
	Copiar prog a la memoria	,	
33. ¿Qué es una variable de condi			
O Varialbe de sincronización que implementa	Variable de sincroniza- ción que permite la espe-		
yield	ra eficiente de un hilo Variable de sincroniza-		
Objeto de sincronización que implementa yield	ción que permite la ex-		
en modo usuario	clusión mutua		
34. ¿Qué es un pipe?)		
Es un buffer del kernel entre dos descriptores de	Es un buffer del kernel entre dos procesos		
archivos	C Es un buffer de usuario		
Es un buffer de usuario entre dos procesos	entre dos descriptores de archivos		
35. ¿Qué son las direcciones de m	nemoria virtuales?		
Memoria que reserva el sistema operativo para el	Capa adicional que utiliza el sistema operativo		
uso de máquinas virtua-	en modo kernel		
les	Capa de indirección que		
Memoria adicional que crea el sistema operativo	le da flexibilidad al siste- ma operativo para admi-		
para engañar a las apli- caciones	nistrar la memoria		



```
1
   char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
10
11
      foo();
12
```

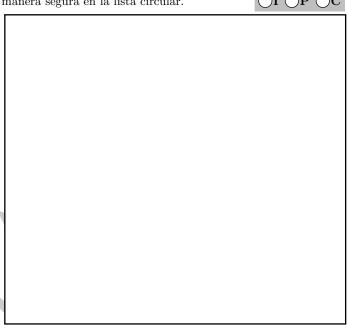
Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse a la dirección de cada instrucción.



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de manera segura en la lista circular. OI OP OC

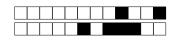


El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

43.	Reescriba la función	Queue::remove	para que elimine de
man	era segura en la lista c	ircular.	$\bigcirc I \ \bigcirc P \ \bigcirc C$





cuestionamientos en las hojas que se le entregan marcando (las respuestas en blanco no se toman en cuenta). <i>Las últim</i>	salirse de ellas (por ejemplo •). Responda a los siguientes o una única opción. Se utilizará factor de corrección 4 a 1 as tres preguntas se evaluarán de manera Completa (4 pts.), e estas últimas tres preguntas, sino que responda en el espacio
	— Marque su RUT sin código verifi- cador (el número después del guión), y escriba sus nombres y apellidos abajo.
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	Nombre(s) y apellido(s):
99999999999	6. ¿Cuántos procesos son creados durante la ejecución del si-
ın evento que requiere su atención. Señal de software Trampa Señal de I/O	<pre>guiente programa? main(int argc, char** argv){ forkthem(5);</pre>
2. Modo de ejecución del procesador en el que verifica cada nstrucción antes de ejecutarla. \(\begin{align*} \text{Modo de usuario} & \text{Modo seguro} \\ \text{Modo dual} & \text{Modo dual} \end{align*}	<pre>} void forkthem(int n){ if (n > 0){ fork(); forkthem(n-1);</pre>
3. Región de la memoria reservada por el sistema operativo para mantener el estado de las variables locales durante la llamada a funciones.	$ \begin{array}{cccc} 5 & \bigcirc 4 \\ \bigcirc 30 & \bigcirc 16 \end{array} $
○ Stack○ Heap○ RAM	7. En la creación del proceso del programa prog, ¿qué paso es el siguiente después de inicializar el espacio de memoria?
4. ¿Qué es la ejecución de una aplicación con permisos restringidos? Aplicación Proceso Programa	 Informar al calendarizador que el programa se puede ejecutar Copiar los argumentos en la memoria Inicializar el hardware para que se ejecute el incio del programa Cargar prog en la memoria
5. ¿Donde almacena el sistema operativo la toda la información sobre un proceso en particular? O Stack O Heap O Memoria	8. ¿Qué son las direcciones de memoria virtuales? O Memoria que reserva el sistema operativo para el uso de máquinas virtuales O Capa adicional que utiliza el sistema operativo en modo kernel O Capa adicional que utiliza el sistema operativo en modo kernel O Capa adicional que utiliza el sistema operativo para engañar a las aplicaciones
	9. ¿Qué es un hilo? Varias secuencias de ejecución que pueden ser calendarizadas independientemente Una secuencia de ejecución que es calendarizada Una secuencia de ejecución que puedes er calendarizada Varias secuencias de ejecución que puedes er calendarizada

dependientemente

den ser calendarizadas dependientemente



10. ¿Cuándo se da una condiciónVarios hilos se pueden intercalar en su ejecución	de carrera? Varios hilos se ejecutan simultáneamente	<pre>main(int argc, char** argv){ int child == fork(); int x = 5;</pre>
simultánea C El estado de la ejecución	Existe una sección crítica	<pre>if (child == 0) { x += 5;</pre>
de un programa depen- de del intercalado de di- ferentes hilos		<pre>} else { child = fork(); x += 10; }</pre>
	rada nor al gistama anarativa	<pre>if (child) { x+=5;</pre>
11. Región de la memoria reser para alojar estructuras de datos q		} }
O Swap	\bigcirc RAM	}
○ Stack	O Heap	19. ¿Cuántas copias distintas de la variable x existen en la eje-
12. ¿Qué hace la función wait de	e una variable de condición?	cución del programa anterior?
Atómicamente libera los seguros y mueve el hilo a la lista de espera	O Duerme el hilo atómica- mente mientras despier- ta al siguiente	$ \bigcirc \begin{array}{c} 4 \\ \bigcirc 3 \end{array} \qquad \bigcirc \begin{array}{c} 2 \\ \bigcirc 1 \end{array} $
Clibera el seguro global, y duerme al hilo	Mueve el hilo a la lista de espera atómicamente	20. ¿Cuál de los siguientes no es un valor que toma la variable x al término de los procesos del programa anterior?
13. ¿Qué es una operación atómi	ica?	$ \begin{array}{c} \bigcirc 5 \\ \bigcirc 15 \end{array} $ $ \begin{array}{c} \bigcirc 10 $
Es una operación indivi-	Es una operación que	
sible que no puede ser di- vidida	deshabilita las interrup- ciones	21. ¿Cuál de las siguientes es una característica de fork? Copiar a la memoria el Crear una copia del pro-
O Es una operación de	O Es una operación del ker-	código del programa del ceso padre, pero no pue-
hardware que no puede ser dividida	nel que no puede ser di- vidida	proceso padre de ser confiado igual que
14. ¿Cuál de los siguientes evento modo kernel a usuario?	os no genera una transición de	Copiar el proceso padre completamente el Copiar el proceso padre con privilegios distintos
Cambiar a un proceso di-	O Llamada I/O a un dispo-	22. ¿Qué es una llamada de sistema?
ferente Continuar después de una interrupción	sitivo O Creación de nuevo proceso	C Llamada que hace el ker- nel para realizar una ins- trucción Funciones de hardware que llama el usuario Funciones del kernel que
15. ¿Qué característica del sistem		Código del kernel que permiten al usuario acce-
independientemente del hardware Portabilidad	Compartimiento de re-	ejecuta codigo de usuario sar a recursos restringi- dos
○ Ilusionista	cursos	23. ¿Qué es el multi-hilado preventivo?
	(Virtualización	Es cuando los hilos que Es cuando los hilos que
16. Es una señal síncrona al produn evento que requiere su atención		se ejecutan pueden ser se ejecutan no pueden cambiados indistinta- descalendarizarse
Señal de software	O Señal de I/O	mente (Es cuando los hilos en-
O Interrupción	Trampa	O Es cuando los hilos no usan el procesador hasta que se les entrega indefi-
17. ¿Qué es la exclusión mutua?		nidamente
O Propiedad de la concu- rrencia para mantener	Objeto de sincronización que bloquea los hilos	24. ¿Qué es un sistema operativo?
los hilos excluidos	Propiedad donde los hi-	C Es un software que admi- No es software, pero se
O Propiedad en la que sola-	los se bloquean mutua-	nistra usuarios encarga de administrar Es una capa de software recursos y usuarios
mente un hilo puede accesar simultáneamente	mente	que administra recursos y usuarios Es un software que administra recursos y usuarios
18. Tipo de kernel en el que la fur de él, y el resto se encuentra a niv		25. ¿Qué es concurrencia?
Híbrido	Macrokernel	Realizar múltiples activi- Realizar una actividad
Monolítico	Microkernel	dades simultáneamente sin la interrupción de Realizar múltiples activi- otra
Dado el siguiente código		dades una después de la Calizar múltiples actividades sin la interrupción de otras



26. ¿Qué hace la función yield de Ouerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting)	e un hilo? Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready)	35. Llamada del sistema de UNIX ciones el comunicarse entre sí para resumirlas. O signal				
Entrega voluntariamente el resto del ciclo en el procesador, y cambia el	Entrega voluntariamente el resto del ciclo en el procesador, y cambia el	∫ fork				
estado a terminado $(fi-nished)$	estado a esperando ($waiting$)	○ Código○ Heap	Variables globalesStack			
27. ¿Cuál de las siguientes no es uTiene un contador de programa propioTiene un stack propio	ina característica de un hilo? Tiene un segmento de datos propio Comparte el código del programa	37. ¿Cuál de las siguientes afirmac spin wait? © Es justificable en monoprocesadores © Es justificable en multi-				
28. ¿Cuál es el principal problema o		procesadores				
No podemos tener varios hilos por procesoNo podemos calendari- zar hilos dentro del pro-	 El bloqueo del proceso, bloquea todos los hilos No podemos tener varios procesos por hilo	38. En la creación del proceso del el siguiente después de cargar prog O Inicializar el hardware para que se ejecute el in-				
ceso		cio del programa	puede ejecutar			
29. ¿Cuál de las siguientes no es criptor de archivo de UNIX?		O Inicializar el PCB en el kernel	Copiar los argumentos en la memoria			
Se abren antes de usarSe cierran completamente después de usar	Se escriben a través de bytesSe leen a través de un buffer	39. ¿Qué es un pipe? Es un buffer de usuario entre dos descriptores de archivos	Es un buffer de usuario entre dos procesos			
30. Tipo de kernel en el que la méste se encuentra dentro de él.	ayoría de la funcionalidad de	Es un buffer del kernel entre dos procesos	Es un buffer del kerne entre dos descriptores de archivos			
O Híbrido	O Macrokernel	40. ¿Cuál de los siguientes elemen	atos no pertenece al <i>TCB</i> ?			
() Microkernel	O Monolítico	O Puntero al stack frame	O Registros salvados			
31. ¿Qué es una variable de condic Objeto de sincronización que implementa yield en modo usuario Varialbe de sincroni- zación que implementa yield	ción? Variable de sincronización que permite la espera eficiente de un hilo Variable de sincronización que permite la exclusión mutua	C Estado del hilo	O Variables locales			
32. ¿Cuál es el estado de un hilo e tarse pero no está en el procesador						
○ Esperando (wait)○ Listo (ready)	$ \bigcirc \ \text{Ejecutandose} \ (\textit{running}) \\ \bigcirc \ \text{Inicializado} \ (\textit{init}) $					
33. ¿Cuál es el primer paso en la grama prog?	creación del proceso del pro-					
 Inicializar el hardware para que se ejecute el proceso desde el inicio Crear e inicializar el PCB en el kernel 	Inicializar la memoriaCopiar prog a la memoria					
34. Modo de ejecución del procesa guna verificación.	ador donde no se ejecuta nin-					
Modo de usuario Modo inseguro	○ Modo kernel○ Modo dual					



```
1
   char bar(){
2
     int b = 2;
3
      return 'a';
4 }
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

$\overline{}$		_	_	_	_		

Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de \bigcirc I \bigcirc P \bigcirc C manera segura en la lista circular.

ı	
1	
1	
1	
ı	
-1	
-1	
-1	
-1	
1	
1	
1	
1	
1	
1	
1	
1	
m	
п	
٧	
1	
1	
1	
1	
ı	
1	
1	
ч	
п	
п	
ų.	
1	
1	

43. Reescriba la función Queue::remove para que elimine de

manera segura en la lista circular.	$\bigcirc I \bigcirc P \bigcirc C$



Instrucciones. Marque las casillas (○) completamente sin cuestionamientos en las hojas que se le entregan marcando (las respuestas en blanco no se toman en cuenta). Las últim Parcial (2 pts.), e Incompleta (0 pts.). No llene las casillas de indicado.	o una única opción. Se utilizar as tres preguntas se evaluarán d	rá factor de corrección 4 a 1 e manera Completa (4 pts.),
$igcircle{0} igcircle{0} igcolumn{0} ig$	Marque su RUT sin c cador (el número después de escriba sus nombres y apelli	el guión), y
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	Nombre(s) y apellido(s):	
1. ¿Cuál de las siguientes es una característica de fork? Crear una copia del proceso padre, pero no puede ser confiado igual que él Copiar el proceso padre completamente Cuál de las siguientes es una característica de fork? Copiar a la memoria el código del programa del proceso padre con privilegios distintos completamente Copiar el proceso padre con privilegios distintos completamente	6. En la creación del proceso del el siguiente después de cargar pro O Inicializar el PCB en el kernel O Copiar los argumentos en la memoria	
 Tiene un segmento de datos propio Tiene un contador de programa propio Comparte el código del programa 	7. ¿Qué es la ejecución de una agidos? O Programa Hilo	plicación con permisos restrin- Aplicación Proceso
3. ¿Cuál es el estado de un hilo en el que está listo para ejecutarse pero no está en el procesador? Listo (ready) Esperando (wait) Ejecutandose (running) Inicializado (init) 4. ¿Cuál de las siguientes afirmaciones es verdadera respecto a spin wait?	8. ¿Cuál es el principal problema No podemos tener varios hilos por proceso No podemos calendari- zar hilos dentro del pro- ceso	
 Es justificable en monoprocesadores No se debe de usar nunca Produce deadlocks Es justificable en multiprocesadores 	9. Modo de ejecución del procesad verificación. Modo inseguro Modo kernel	or donde no se ejecuta ninguna Modo dual Modo de usuario
5. ¿Cuántos procesos son creados durante la ejecución del si- guiente programa?	10. ¿Donde almacena el sistema o sobre un proceso en particular?	perativo la toda la información
<pre>main(int argc, char** argv){ forkthem(5); } void forkthem(int n){ if (n > 0){</pre>		O Stack O Memoria
<pre>if (n > 0){ fork(); forkthem(n-1); } </pre>	11. Es una señal síncrona al produn evento que requiere su atención Interrupción Señal de I/O	
\bigcirc 16 \bigcirc 30		



Dado el siguiente código

THOSE IN SERVER IN		
 ¿Cuándo se da una condición Varios hilos se pueden intercalar en su ejecución simultánea Varios hilos se ejecutan simultáneamente 	de carrera? Existe una sección crítica El estado de la ejecución de un programa depende del intercalado de diferentes hilos	<pre>main(int argc, char** argv){ int child == fork(); int x = 5; if (child == 0) { x += 5; } else { child = fork(); x += 10; }</pre>
 Tipo de kernel en el que la r éste se encuentra dentro de él. 	nayoría de la funcionalidad de	<pre>if (child) { x+=5; </pre>
O Macrokernel	○ Híbrido	, } '
○ Monolítico	○ Microkernel	}
14. Llamada del sistema de UNI ciones el comunicarse entre sí par resumirlas.	a terminarlas, suspenderlas, o	21. ¿Cuál de los siguientes no es un valor que toma la variable x al término de los procesos del programa anterior? 5 10 10
○ wait	<pre>exec</pre>	22. ¿Cuántas copias distintas de la variable x existen en la eje-
15. Región de la memoria reser para alojar estructuras de datos qSwap	= = = = = = = = = = = = = = = = = = = =	cución del programa anterior?
○ RAM	O Heap	23. ¿Qué es una operación atómica?
16. ¿Cuál es el primer paso en la grama prog? Crear e inicializar el PCB en el kernel Inicializar el hardware para que se ejecute el	Copiar prog a la memoria Inicializar la memoria	 Es una operación que deshabilita las interrupciones Es una operación indivisible que no puede ser dividida Es una operación del kernel que no puede ser dividida Es una operación del kernel que no puede ser dividida Es una operación indivisible que no puede ser dividida Es una operación indivisible que no puede ser dividida Es una operación indivisible que no puede ser dividida Es una operación indivisible que no puede ser dividida
proceso desde el inicio 17. ¿Qué característica del sistem independientemente del hardware Portabilidad Virtualización		24. ¿Qué hace la función wait de una variable de condición? Atómicamente libera los seguros y mueve el hilo a mente mientras despierta al siguiente Libera el seguro global, y duerme al hilo sepera atómicamente
 18. En la creación del proceso de el siguiente después de inicializar Copiar los argumentos en la memoria Informar al calendarizador que el programa se 	cursos l programa prog , ¿qué paso es	25. ¿Qué es una llamada de sistema? Clamada que hace el kernel para realizar una instrucción Funciones del kernel que permiten al usuario accesar a recursos restringi-
puede ejecutar 19. Región de la memoria reservad mantener el estado de las variable funciones.		dos 26. ¿Qué son las direcciones de memoria virtuales? Capa de indirección que Capa adicional que utile da flexibilidad al sistema operativo
○ Heap○ Swap20. ¿Qué es concurrencia?	O RAM O Stack	ma operativo para administrar la memoria Memoria que reserva el sistema operativo para el uso de máquinas virtua- en modo kernel Memoria adicional que crea el sistema operativo para engañar a las aplicaciones
Realizar múltiples actividades sin la interrupción de otras	Realizar múltiples actividades una después de la otra	les ${\bf 27.} \ \ {\it i_C} {\it Cu\'al} \ {\it de las siguientes no es una caracter\'atica de un des-}$
Realizar una actividad sin la interrupción de otra	O Realizar múltiples actividades simultáneamente	criptor de archivo de UNIX? Se abren antes de usar Se cierran completamen- Se escriben a través de bytes

O Se leen a través de un

buffer

te después de usar



udp Escuela de Informática
y Telecomunicaciones
FACULTAD DE INGENIERÍA

90 M 1 1 ' '/ 11	1 1 :0 1	97 .C.(1.1.1	1 / 1 / 1 / 1 / 1 / 1 / 1 / 1 / 1 / 1 /
28. Modo de ejecución del pro- instrucción antes de ejecutarla.	cesador en el que verifica cada	37. ¿Cuál de los siguientes elementos del hilo	Puntero al stack frame
O Modo seguro	O Modo kernel	Registros salvados	Variables locales
O Modo de usuario	O Modo dual	9	
29. ¿Cuál de los siguientes event	tos no genera una transición de	38. Tipo de kernel en el que la fu de él, y el resto se encuentra a ni	
modo kernel a usuario? Clamada I/O a un dispo-	Creación de nuevo proce-	O Híbrido	Microkernel
sitivo	So	○ Monolítico	(Macrokernel
Cambiar a un proceso di-	O Continuar después de	39. ¿Qué es un hilo?	
ferente	una interrupción	O Una secuencia de ejecu-	O Una secuencia de ejecu-
30. Es una señal asíncrona al pr		ción que es calendarizada dependientemente	ción que puede ser calen- darizada independiente-
un evento que requiere su atenció		Varias secuencias de eje-	mente
○ Señal de I/O○ Señal de software	○ Trampa○ Interrupción	cución que pueden ser	Varias secuencias de
	_	calendarizadas indepen- dientemente	ejecución que pue- den ser calendarizadas
31. ¿Qué es la exclusión mutua?		dientemente	dependientemente
O Propiedad de la concurrencia para mantener	O Propiedad en la que sola- mente un hilo puede ac-	40. ¿Qué es el multi-hilado prev	entivo?
los hilos excluidos	cesar simultáneamente	Es cuando los hilos que	Es cuando los hilos no
O Propiedad donde los hi-	Objeto de sincronización	se ejecutan pueden ser	usan el procesador hasta
los se bloquean mutua- mente	que bloquea los hilos	cambiados indistinta- mente	que se les entrega indefi- nidamente
32. ¿Qué es una variable de con	dición?	O Es cuando los hilos en-	Es cuando los hilos que
Objeto de sincronización	O Varialbe de sincroni-	tregan el procesador vo-	se ejecutan no pueden descalendarizarse
que implementa yield	zación que implementa	luntariamente, no son in- terrumpidos	descalendarizarse
en modo usuario Variable de sincroniza-	yield O Variable de sincroniza-	7	
ción que permite la ex-	ción que permite la espe-		
clusión mutua	ra eficiente de un hilo		
33. ¿Cuál de los siguientes elem hilos?	entos no es compartido por los		
○ Código	○ Stack		
O Variables globales	Heap		
34. ¿Qué es un sistema operativ	70?		
Es un software que admi-	C Es un software que admi-		
nistra recursos	nistra usuarios		
Es una capa de software que administra recursos	O No es software, pero se encarga de administrar		
y usuarios	recursos y usuarios		
35. ¿Qué es un pipe?			
O Es un buffer del kernel	O Es un buffer de usuario		
entre dos procesos	entre dos descriptores de archivos		
() Es un buffer del kernel entre dos descriptores de	Es un buffer de usuario		
archivos	entre dos procesos		
36. ¿Qué hace la función yield	de un hilo?		
Entrega voluntariamente	Entrega voluntariamente		
el resto del ciclo en el procesador, y cambia el	el resto del ciclo en el procesador, y cambia el		
estado a terminado (fi-	estado a esperando (wai -		
nished)	ting)		
O Entrega voluntariamente el resto del ciclo en el	O Duerme al hilo por un tiempo determinado, y		
procesador, y cambia el	cambia el estado a espe-		
estado a listo $(ready)$	rando(waiting)		



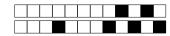
```
1
   char bar(){
2
     int b = 2;
3
      return 'a';
4 }
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
10
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcirc I \bigcirc D \bigcirc C$ manara sagura en la lista siraular

manera segura en la lista circular.	$\bigcirc 1 \bigcirc P \bigcirc C$

43. Reescriba la función Queue::remove para que elimine de

manera segura en la lista circular.	\bigcirc I \bigcirc P \bigcirc C



Instrucciones. Marque las casillas (\bigcirc) completamente sin cuestionamientos en las hojas que se le entregan marcando (las respuestas en blanco no se toman en cuenta). Las último Parcial (2 pts.), e Incompleta (0 pts.). No llene las casillas de indicado.	una única opción. Se utilizará factor de corrección 4 a 1 as tres preguntas se evaluarán de manera Completa (4 pts.),
	← Marque su RUT sin código verificador (el número después del guión), y escriba sus nombres y apellidos abajo. Nombre(s) y apellido(s):
1. ¿Cuál es el estado de un hilo en el que está listo para ejecutarse pero no está en el procesador?	7. ¿Cuál de las siguientes afirmaciones es verdadera respecto a spin wait?
	Es justificable en mono- procesadores Es justificable en multi- procesadores
2. ¿Qué es un hilo?	O Produce deadlocks O No se debe de usar nunca
Varias secuencias de ejecución que pueden ser calendarizadas independientemente Una secuencia de ejecución que puedes er calendarizada independientemente Una secuencia de ejecución que puedes er calendarizada dependientemente Una secuencia de ejecución que es calendarizada dependientemente	8. ¿Qué es el multi-hilado preventivo? Es cuando los hilos no usan el procesador hasta que se les entrega indefinidamente Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente
3. ¿Cuál de las siguientes no es una característica de un descriptor de archivo de UNIX?	9. ¿Qué característica del sistema operativo permite que opere independientemente del hardware en la computadora?
 Se abren antes de usar Se escriben a través de bytes Se cierran completamente después de usar Se leen a través de un buffer 	 ○ Virtualización ○ Compartimiento de recursos ○ Portabilidad ○ Ilusionista
4. Es una señal asíncrona al procesador que indica que ocurrió un evento que requiere su atención.	${\bf 10.}\;\; {\bf ¿}$ Qué es la ejecución de una aplicación con permisos restringidos?
○ Señal de software○ Señal de I/O○ Trampa	○ Aplicación○ Proceso○ Hilo○ Programa
5. Modo de ejecución del procesador donde no se ejecuta ninguna verificación.	11. ¿Cuál de los siguientes elementos ${f no}$ es compartido por los hilos?
O Modo de usuario O Modo kernel	○ Código ○ Heap
○ Modo inseguro ○ Modo dual	○ Variables globales ○ Stack
6. En la creación del proceso del programa prog, ¿qué paso es el siguiente después de inicializar el espacio de memoria? Copiar los argumentos O Inicializar el hardware para que se ejecute el incio del programa cio del programa Informar al calendarizador que el programa se puede ejecutar	12. Región de la memoria reservada por el sistema operativo para mantener el estado de las variables locales durante la llamada a funciones. Swap Stack RAM



13. ¿Cuál es el primer paso en la creación del proceso del programa prog? Crear e inicializar el Copiar prog a la memoria Inicializar el hardware para que se ejecute el proceso desde el inicio 14. ¿Cuántos procesos son creados durante la ejecución del si-	 ○ Propiedad en la que solamente un hilo puede accesar simultáneamente ○ Propiedad donde los hilos se bloquean mutuamente ○ Propiedad donde los hilos se bloquean mutuamente ○ Objeto de sincronización que bloquea los hilos 22. ¿Cuál de las siguientes no es una característica de un hilo?
<pre>guiente programa? main(int argc, char** argv){ forkthem(5); }</pre>	 ○ Tiene un stack propio ○ Comparte el código del programa ○ Tiene un contador de programa propio ○ Tiene un segmento de datos propio
void forkthem(int n){ if (n > 0){ fork(); forkthem(n-1); } } 4	23. ¿Cuándo se da una condición de carrera? Existe una sección crítica Simultáneamente Varios hilos se pueden intercalar en su ejecución de un programa depende del intercalado de diferentes hilos Dado el siguiente código main (int argc, char** argv) { int child == fork(); int x = 5; if (child == 0) { x += 5; } else { child = fork(); x += 10;
16. ¿Donde almacena el sistema operativo la toda la información sobre un proceso en particular? Memoria Process control block Heap	<pre>if (child) { x+=5; } }</pre>
(PCB) 17. Tipo de kernel en el que la mayoría de la funcionalidad de este se encuentra dentro de él.	24. ¿Cuántas copias distintas de la variable x existen en la ejecución del programa anterior?
 Monolítico Macrokernel Microkernel	$ \bigcirc \ 4 \qquad \qquad \bigcirc \ 2 \\ \bigcirc \ 1 \qquad \qquad \bigcirc \ 3 $
18. Es una señal síncrona al procesador que indica que ocurrió un evento que requiere su atención.	25. ¿Cuál de los siguientes no es un valor que toma la variable x al término de los procesos del programa anterior?
☐ Interrupción ☐ Señal de I/O ☐ Señal de software ☐ Trampa	$ \begin{array}{ccc} \bigcirc 10 & \bigcirc 20 \\ \bigcirc 15 & \bigcirc 5 \end{array} $
19. ¿Cuál de las siguientes es una característica de fork? Crear una copia del proceso padre completamente de ser confiado igual que él Copiar el proceso padre con privilegios distintos Cupiar el proceso padre con privilegios distintos	26. ¿Qué es una variable de condición? Variable de sincronización que permite la espera eficiente de un hilo Variable de sincronización que permite la exclusión que permite la exclusión mutua 27. ¿Qué hace la función wait de una variable de sincronización?
20. ¿Cuál es el principal problema de los hilos a nivel de usuario? No podemos calendari- zar hilos dentro del pro- bloquea todos los hilos	Mueve el hilo a la lista de espera atómicamente duerme al hilo

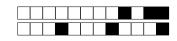


28. Región de la memoria reservar alojar estructuras de datos q	ue el proceso pueda necesitar.	37. Llamada del sistema de UNIX que le permite a las aplica ciones el comunicarse entre sí para terminarlas, suspenderlas, resumirlas.
○ Swap○ Heap	○ Stack ○ RAM	○ wait ○ fork
 29. ¿Cuál de los siguientes evento modo kernel a usuario? Llamada I/O a un dispositivo Cambiar a un proceso diferente 30. Modo de ejecución del proceinstrucción antes de ejecutarla. Modo seguro Modo de usuario 	 Continuar después de una interrupción Creación de nuevo proceso 	38. ¿Qué hace la función yield de un hilo? Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (waiting) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (fi-
 31. ¿Qué es una operación atómico Es una operación del kernel que no puede ser dividida Es una operación de hardware que no puede ser dividida 	 Es una operación que deshabilita las interrupciones Es una operación indivisible que no puede ser dividida 	nished) 39. ¿Cuál de los siguientes elementos no pertenece al TCB? \[\begin{align*} \text{Variables locales} & \text{\text{Descendents}} \\ \text{Registros salvados} & \text{\text{Puntero al stack frame}} \\ 40. ¿Qué son las direcciones de memoria virtuales? \[\begin{align*} \text{Capa adicional que uti-} & \text{Memoria adicional que liza el sistema operativo} \\ \end{align*} \text{Crea el sistema operativo} \\ \end{align*}
 ¿Qué es una llamada de siste Funciones del kernel que permiten al usuario accesar a recursos restringidos Código del kernel que ejecuta código de usuario 	Ma? Llamada que hace el kernel para realizar una instrucción Funciones de hardware que llama el usuario	en modo kernel Memoria que reserva el sistema operativo para el uso de máquinas virtuales Crea el sistema operativo para el caciones Capa de indirección que le da flexibilidad al siste ma operativo para administrar la memoria
 33. ¿Qué es un sistema operativo Es un software que administra usuarios Es una capa de software que administra recursos y usuarios 	O? One software, pero se encarga de administrar recursos y usuarios Es un software que administra recursos	
 34. ¿Qué es un pipe? Es un buffer de usuario entre dos procesos Es un buffer de usuario entre dos descriptores de archivos 	Es un buffer del kernel entre dos procesos Es un buffer del kernel entre dos descriptores de archivos	
 35. En la creación del proceso de el siguiente después de cargar proposition de la procesa de la siguiente después de cargar proposition de la programa Copiar los argumentos en la memoria 36. Tipo de kernel en el que la fun 	g en la memoria? Inicializar el PCB en el kernel Informar al calendarizador que el programa se puede ejecutar cionalidad mínima está dentro	
de él, y el resto se encuentra a niv	el de usuario. Microkernel Macrokernel	



```
1
    char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
      int b=2;
10
11
      foo();
12
```

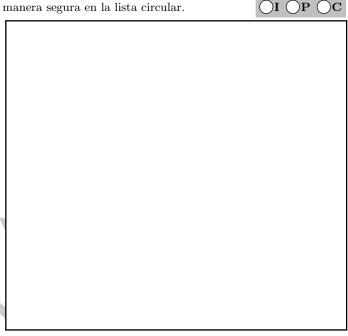
Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse a la dirección de cada instrucción.



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de

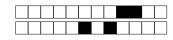


El siguiente código implementa una lista circular.

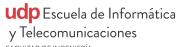
```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

43. Reescriba la función Queue::remove para que elimine de manera segura en la lista circular. $\bigcirc I \bigcirc P \bigcirc C$





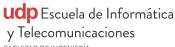
		salirse de ellas (por ejemplo ●)	
		una única opción. Se utilizara	
` -	,	as tres preguntas se evaluarán de estas últimas tres preguntas, sinc	- (:-):
indicado.	pis.). No tiene las castilas de	estas utilmas tres preguntas, sino	que responda en el espacio
$\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0$	$\bigcirc 0 \bigcirc 0 \bigcirc 0 \bigcirc 0$	N. DIE	/ 1
		← Marque su RUT sin co	
$\bigcirc 1\bigcirc 1\bigcirc$		cador (el número después de	
$\bigcirc 2\bigcirc 2\bigcirc$		escriba sus nombres y apellio	ios abajo.
$\bigcirc 3\bigcirc 3\bigcirc$		NT 1 () 11:1 ()	
$\bigcirc 4\bigcirc 4\bigcirc$		Nombre(s) y apellido(s):	
$\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc$	$\bigcirc 5 \bigcirc 5 \bigcirc 5 \bigcirc 5$		
$\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc$	$\bigcirc 6\bigcirc 6\bigcirc 6\bigcirc 6$		
$\bigcirc 7 \bigcirc 7 \bigcirc 7 \bigcirc 7$	$7 \bigcirc 7 \bigcirc 7 \bigcirc 7$		
$\bigcirc 9 \bigcirc 9 \bigcirc 9 \bigcirc 9 \bigcirc 9$			
1. ¿Qué es la ejecución de una ap	licación con permisos restrin-	7. ¿Cuándo se da una condición d	
gidos?		Existe una sección críti-	Varios hilos se pueden in-
O Aplicación	(Hilo	Ca Varios biles as sissuter	tercalar en su ejecución simultánea
O Proceso	O Programa	Varios hilos se ejecutan simultáneamente	El estado de la ejecución
2. ¿Cuál de las siguientes no es	una característica de un des-	Simulatedicalicitie	de un programa depen-
criptor de archivo de UNIX?	ana caracteristica de un des		de del intercalado de di-
Se escriben a través de	O Se leen a través de un		ferentes hilos
bytes	buffer	8. ¿Donde almacena el sistema op	erativo la toda la información
O Se cierran completamen-	O Se abren antes de usar	sobre un proceso en particular?	
te después de usar		() Stack	(Memoria
3. Tipo de kernel en el que la ma	avoría de la funcionalidad de		O Process control block
éste se encuentra dentro de él.			(PCB)
O Monolítico	(Macrokernel	0. I lamada dal gistama da UNIV	que le permite e les eplice
(Microkernel	() Híbrido	9. Llamada del sistema de UNIX ciones el comunicarse entre sí para	= = =
	O Illorido	resumirlas.	t terminarias, suspenderias, e
4. ¿Qué es la exclusión mutua?		exec	O wait
O Propiedad en la que sola-	O Propiedad donde los hi-	Signal	o fork
mente un hilo puede ac-	los se bloquean mutua-	Signal	OTOTA
cesar simultáneamente Objeto de sincronización	mente O Propiedad de la concu-	10. ¿Cuál de las siguientes afirma	ciones es verdadera respecto a
que bloquea los hilos	rrencia para mantener	spin wait?	
1	los hilos excluidos	Es justificable en mono-	O Produce deadlocks
5. ¿Cuál es el estado de un hilo en e	l que está listo para ejecutarse	procesadores	O No se debe de usar nunca
pero no está en el procesador?	i que esta listo para ejecutarse	Es justificable en multi- procesadores	
○ Inicializado (init)	C Ejecutandose (running)	procesadores	
Esperando (wait)	Listo (ready)	11. Modo de ejecución del proces	ador donde no se ejecuta nin-
C Esperando (waii)	C Listo (ready)	guna verificación.	
6. Modo de ejecución del proces	ador en el que verifica cada	Modo kernel	Modo de usuario
instrucción antes de ejecutarla.		Modo inseguro	Modo dual
O Modo dual	O Modo kernel	12. ¿Cuál es el principal problema	de los hilos a nivel de usuario?
O Modo de usuario	O Modo seguro	No podemos tener varios	El bloqueo del proceso,
		procesos por hilo	bloquea todos los hilos
		O No podemos calendari-	O No podemos tener varios
		zar hilos dentro del pro-	hilos por proceso
		ceso	



ΕΔΟΙΙΙΤΔΩ	DE	INGENIERÍA	

40 0 () ! ! !	1 9	24 5 21 ()	
13. ¿Qué son las direcciones de modernes de indirección que	emoria virtuales? Memoria adicional que	21. Es una señal síncrona al procun evento que requiere su atención	
le da flexibilidad al siste-	crea el sistema operativo	Interrupción	Señal de software
ma operativo para admi-	para engañar a las apli-	(Trampa	Señal de I/O
nistrar la memoria	caciones		,
Memoria que reserva el sistema operativo para el	Capa adicional que utiliza el sistema operativo	22. ¿Cuántos procesos son creado guiente programa?	os durante la ejecución del si-
uso de máquinas virtua-	en modo kernel	guiente programa:	
les		main(int argc, o	har** argv){
14. ¿Qué es una llamada de sisten	na?	<pre>forkthem(5); }</pre>	
Código del kernel que	Funciones de hardware	void forkthem(in	t n){
ejecuta código de usuario Llamada que hace el ker-	que llama el usuario Funciones del kernel que	<pre>if (n > 0){ fork();</pre>	
nel para realizar una ins-	permiten al usuario acce-	forkthem(n-1));
trucción	sar a recursos restringi-	}	
	dos	\bigcirc 30	\bigcirc 16
15. Región de la memoria reserva		\bigcirc 50 \bigcirc 5	\bigcirc 4
para alojar estructuras de datos qu	<u> </u>		
○ Stack	Swap	23. ¿Qué es un sistema operativo	
() Heap	() RAM	Es un software que administra usuarios	Es una capa de software que administra recursos
16. ¿Cuál de las siguientes no es u	_	No es software, pero se	y usuarios
Tiene un stack propio	Tiene un contador de	encarga de administrar	C Es un software que admi-
Oceanie Comparte el código del	programa propio Tiene un segmento de	recursos y usuarios	nistra recursos
programa	datos propio	24. ¿Qué hace la función wait de	
17. Tipo de kernel en el que la func	ionalidad mínima está dentro	 Mueve el hilo a la lista de espera atómicamente 	Atómicamente libera los seguros y mueve el hilo a
de él, y el resto se encuentra a nive		Libera el seguro global, y	la lista de espera
Microkernel	(Macrokernel	duerme al hilo	O Duerme el hilo atómica-
◯ Híbrido	O Monolítico		mente mientras despier-
18. ¿Qué hace la función yield de	un hilo?	,	ta al siguiente
Entrega voluntariamente	Duerme al hilo por un	Dado el siguiente código	
el resto del ciclo en el	tiempo determinado, y	main(int argc, o	
procesador, y cambia el	cambia el estado a espe-	<pre>int child == f int x = 5;</pre>	fork();
estado a esperando ($wai-ting$)	rando (waiting)	if (child == 0) {
Entrega voluntariamente	O Entrega voluntariamente el resto del ciclo en el	x += 5; } else {	
el resto del ciclo en el	procesador, y cambia el	child = fork	();
procesador, y cambia el	estado a listo $(ready)$	x += 10; if (child) {	
estado a terminado (fi- nished))	x+=5;	•
19. En la creación del proceso del	, , , , , , , , , , , , , , , , , , ,	}	
el siguiente después de cargar prog		}	
Informar al calendariza-	O Inicializar el hardware	25. ¿Cuál de los siguientes no es	un valor que toma la variable
dor que el programa se	para que se ejecute el in-	x al término de los procesos del pr	ograma anterior?
puede ejecutar	cio del programa	\bigcirc 5	\bigcirc 15
Copiar los argumentos en la memoria	O Inicializar el PCB en el kernel	\bigcirc 20	O 10
20. En la creación del proceso del		26. ¿Cuántas copias distintas de	la variable x existen en la eje-
el siguiente después de inicializar el		cución del programa anterior?	-
O Informar al calendariza-	O Inicializar el hardware	\bigcirc 2	\bigcirc 3
dor que el programa se	para que se ejecute el in-	\bigcirc 1	\bigcirc 4
puede ejecutar Cargar prog en la memo-	cio del programa Copiar los argumentos	27. ¿Cuál de los siguientes eleme	ntos no es compartido por los
ria	en la memoria	hilos?	- •
		O Stack	O Variables globales
		○ Código	Heap

Para revisión, preferentemente imprimir documentos compilados desde automultiple-choice.



UOP Escuela de Informática
y Telecomunicaciones
FACULTAD DE INGENIERÍA

28. Es una señal asíncrona al produn evento que requiere su atención		36. ¿Qué característica del sistem independientemente del hardware	
O Señal de software	O Trampa	Ocompartimiento de re-	() Ilusionista
○ Señal de I/O	O Interrupción	cursos	O Portabilidad
29. ¿Qué es una operación atómic Es una operación de	ca? © Es una operación que	Virtualización37. ¿Cuál es el primer paso en la	a creación del proceso del pro-
hardware que no puede ser dividida Es una operación del ker-	deshabilita las interrupciones Es una operación indivi-	grama prog? Crear e inicializar el PCB en el kernel	Copiar prog a la memoria
nel que no puede ser di- vidida	sible que no puede ser di- vidida	O Inicializar la memoria	O Inicializar el hardware para que se ejecute el proceso desde el inicio
30. ¿Qué es el multi-hilado prever		38. ¿Cuál de los siguientes elemen	
() Es cuando los hilos en-	Es cuando los hilos que		
tregan el procesador vo- luntariamente, no son in-	se ejecutan no pueden descalendarizarse	Variables localesPuntero al stack frame	Estado del hiloRegistros salvados
terrumpidos	Es cuando los hilos que se ejecutan pueden ser	39. ¿Qué es una variable de cond	ición?
 Es cuando los hilos no usan el procesador hasta que se les entrega indefi- nidamente 	cambiados indistinta- mente	Objeto de sincronización que implementa yield en modo usuario	Varialbe de sincronización que implementa
31. ¿Cuál de los siguientes evento modo kernel a usuario?	s no genera una transición de	O Variable de sincroniza- ción que permite la ex-	Variable de sincroniza- ción que permite la espe-
O Llamada I/O a un dispo-	Creación de nuevo proce-	clusión mutua	ra eficiente de un hilo
sitivo O Continuar después de	Cambiar a un proceso di-	40. ¿Qué es un pipe? Es un buffer del kernel	Es un buffer de usuario
una interrupción	ferente	entre dos procesos	entre dos procesos
32. ¿Cuál de las siguientes es una		Es un buffer del kernel entre dos descriptores de	Es un buffer de usuario entre dos descriptores de
Copiar a la memoria el código del programa del	Copiar el proceso padre completamente	archivos	archivos
proceso padre Copiar el proceso padre	Crear una copia del proceso padre, pero no pue-	>	
con privilegios distintos	de ser confiado igual que él		
99 0 / 111 9	er		
33. ¿Qué es un hilo? Una secuencia de ejecu-	Varias secuencias de eje-		
ción que puede ser calen-	cución que pueden ser		
darizada independiente- mente	calendarizadas indepen- dientemente		
Varias secuencias de	Una secuencia de ejecu-		
ejecución que pue-	ción que es calendarizada		
den ser calendarizadas dependientemente	dependientemente		
34. ¿Qué es concurrencia?			
Realizar múltiples actividades sin la interrupción	Realizar múltiples actividades una después de la		
de otras	otra		
Realizar una actividad sin la interrupción de	Realizar múltiples actividades simultáneamente		
otra			
35. Región de la memoria reservado mantener el estado de las variables funciones.			
Stack	(Heap		
○ RAM	○ Swap		
~			



```
char bar(){
2
     int b = 2;
3
      return 'a';
   }
4
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
      int b=2;
10
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

Г						
Ė	İ					

Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

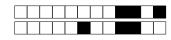
Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcirc I \bigcirc P \bigcirc C$ manera segura en la lista circular.

43. Reescriba la función Queue::remove para que elimine de

manera segura en la lista circular.	\bigcirc I \bigcirc P \bigcirc C





cuestionamientos en las hojas qu (las respuestas en blanco no se t	ue se le entregan marcando coman en cuenta). <i>Las últim</i>	salirse de ellas (por ejemplo una única opción. Se utiliza as tres preguntas se evaluarán a e estas últimas tres preguntas, sin	rá factor de corrección 4 a 1 le manera Completa (4 pts.)
0000000 111111 222222 333333 44444 555555 666666 7777777 888888 9999999	1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 7 7 7 7 8 8 8 8 9 9 9 9 9	← Marque su RUT sin of cador (el número después de escriba sus nombres y apell Nombre(s) y apellido(s): main(int argc,	el guión), y idos abajo. char** argv){
 Varias secuencias de ejecución que pueden ser calendarizadas dependientemente Una secuencia de ejecución que es calendarizada dependientemente 	 Una secuencia de ejecución que puede ser calendarizada independientemente Varias secuencias de ejecución que pueden ser calendarizadas independientemente 	<pre>int child == int x = 5; if (child == x += 5; } else { child = for x += 10; if (child) x+=5; }</pre>	0) { k();
2. ¿Cuál es el estado de un hilo en e pero no está en el procesador?	l que está listo para ejecutarse	, , ,	
○ Ejecutandose (running)○ Esperando (wait)	◯ Listo (ready)◯ Inicializado (init)	6. ¿Cuántas copias distintas de la ción del programa anterior?	a variable \mathbf{x} existen en la ejecu-
3. ¿Qué es concurrencia? Realizar múltiples actividades simultáneamente Realizar una actividad sin la interrupción de otra	Realizar múltiples actividades una después de la otra Realizar múltiples actividades sin la interrupción de otras	1 0 4 7. ¿Cuál de los siguientes no es a al término de los procesos del pro 5 10	
4. Región de la memoria reservada mantener el estado de las variables funciones.	por el sistema operativo para	8. ¿Qué característica del sistem independientemente del hardware	na operativo permite que opere
Swap Stack	○ RAM○ Heap	O Portabilidad Ilusionista	Compartimiento de recursos Virtualización
5. ¿Cuál de los siguientes elementales?	tos no es compartido por los	9. ¿Cuál de los siguientes evento modo kernel a usuario?	os no genera una transición de
Stack	Código	Ulamada I/O a un dispositivo	Creación de nuevo proce-
HeapDado el siguiente código	Variables globales	Continuar después de una interrupción	So Cambiar a un proceso diferente
sade of signification codings		10. ¿Cuál de los siguientes eleme	
		Variables locales	Puntero al stack frame

O Estado del hilo

O Registros salvados



 ¿Cuál es el primer paso en la grama prog? Copiar prog a la memoria Inicializar el hardware para que se ejecute el proceso desde el inicio 	Creación del proceso del pro- Crear e inicializar el PCB en el kernel Inicializar la memoria	 ¿Qué es la exclusión mutua? Propiedad donde los hilos se bloquean mutuamente Propiedad en la que solamente un hilo puede accesar simultáneamente 	 Objeto de sincronización que bloquea los hilos Propiedad de la concurrencia para mantener los hilos excluidos
 12. ¿Cuál de las siguientes no es Tiene un segmento de datos propio Comparte el código del programa 	una característica de un hilo? O Tiene un contador de programa propio O Tiene un stack propio	 21. ¿Cuál de las siguientes afirma spin wait? Produce deadlocks Es justificable en multiprocesadores 	Ciones es verdadera respecto a Es justificable en monoprocesadores No se debe de usar nunca
 13. En la creación del proceso de el siguiente después de cargar proceso de la siguiente después de cargar proceso de la siguiente después de cargar proceso de la siguiente del siguiente de la sigu	g en la memoria? Informar al calendarizador que el programa se puede ejecutar Copiar los argumentos en la memoria cionalidad mínima está dentro el de usuario.	<pre>22. ¿Cuántos procesos son cread guiente programa? main(int argc, of forkthem(5); } void forkthem(int if (n > 0){ fork(); forkthem(n-1) }</pre>	char** argv){ nt n){
	○ Híbrido○ Macrokernel○ perativo la toda la información) 16 () 30	○ 5○ 4
sobre un proceso en particular? Heap Memoria		23. Es una señal síncrona al produn evento que requiere su atención Trampa Interrupción	
 ¿Qué es una llamada de siste Código del kernel que ejecuta código de usuario Funciones del kernel que permiten al usuario accesar a recursos restringidos 	ma? Llamada que hace el kernel para realizar una instrucción Funciones de hardware que llama el usuario	 24. ¿Qué es una operación atómi ○ Es una operación del kernel que no puede ser dividida ○ Es una operación de hardware que no puede ser dividida 	ca? Es una operación indivisible que no puede ser dividida Es una operación que deshabilita las interrupciones
 17. ¿Cuál de las siguientes no escriptor de archivo de UNIX? Se escriben a través de bytes Se abren antes de usar 18. ¿Cuál de las siguientes es una Copiar el proceso padre con privilegios distintos 	 Se cierran completamente después de usar Se leen a través de un buffer 	 ¿Qué son las direcciones de n Capa de indirección que le da flexibilidad al sistema operativo para administrar la memoria Memoria adicional que crea el sistema operativo para engañar a las aplicaciones 	nemoria virtuales? Capa adicional que utiliza el sistema operativo en modo kernel Memoria que reserva el sistema operativo para el uso de máquinas virtuales
Crear una copia del pro- ceso padre, pero no pue- de ser confiado igual que él	O Copiar a la memoria el código del programa del proceso padre	26. Modo de ejecución del proces guna verificación. Modo dual Modo inseguro	sador donde no se ejecuta nin- Modo kernel Modo de usuario
19. Región de la memoria reservaria alojar estructuras de datos qRAMHeap	*	27. ¿Qué es un sistema operativo Es una capa de software que administra recursos y usuarios Es un software que admi-	

nistra usuarios

recursos y usuarios



28. ¿Qué es un pipe? Es un buffer del kernel entre dos descriptores de archivos Es un buffer de usuario entre dos procesos 29. Llamada del sistema de UNIX ciones el comunicarse entre sí para resumirlas. wait		 36. ¿Qué hace la función yield de Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a terminado (finished) Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a listo (ready) 	e un hilo? Entrega voluntariamente el resto del ciclo en el procesador, y cambia el estado a esperando (waiting) Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting)
fork 30. Tipo de kernel en el que la majéste se encuentra dentro de él. Híbrido Monolítico	exec	 37. ¿Qué hace la función wait de Libera el seguro global, y duerme al hilo Atómicamente libera los seguros y mueve el hilo a la lista de espera 	una variable de condición? Mueve el hilo a la lista de espera atómicamente Duerme el hilo atómicamente mientras despierta al siguiente
 ¿Qué es el multi-hilado prevent Es cuando los hilos entregan el procesador voluntariamente, no son interrumpidos Es cuando los hilos que se ejecutan no pueden descalendarizarse 	ivo? Es cuando los hilos no usan el procesador hasta que se les entrega indefinidamente Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente	 38. ¿Cuándo se da una condición Varios hilos se pueden intercalar en su ejecución simultánea El estado de la ejecución de un programa depende del intercalado de diferentes hilos 39. Es una señal asíncrona al production 	Varios hilos se ejecutan simultáneamente Existe una sección crítica ca
32. ¿Qué es la ejecución de una apligidos? () Hilo	cación con permisos restrin-	un evento que requiere su atención Interrupción Señal de I/O	i. Trampa Señal de software
Programa (33. ¿Cuál es el principal problema de El bloqueo del proceso, bloquea todos los hilos (No podemos tener varios procesos por hilo	Aplicación e los hilos a nivel de usuario? No podemos tener varios hilos por proceso No podemos calendarizar hilos dentro del proceso	40. En la creación del proceso del el siguiente después de inicializar el Cargar prog en la memoria Copiar los argumentos en la memoria	el espacio de memoria? Informar al calendarizador que el programa se puede ejecutar Inicializar el hardware para que se ejecute el in-
34. Modo de ejecución del procesa instrucción antes de ejecutarla. Modo dual Modo seguro 35. ¿Qué es una variable de condición que permite la exclusión mutua Variable de sincronización que permite la espera eficiente de un hilo	ador en el que verifica cada Modo de usuario Modo kernel		cio del programa



```
1
   char bar(){
2
     int b = 2;
3
     return 'a';
4 }
   int foo(int a, char b){
6
      int i=0;
7
      bar();
   }
8
9
   main(){
     int b=2;
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

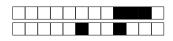
42. Reescriba la función Queue::insert para que inserte de $\bigcirc I \bigcirc P \bigcirc C$ manera segura en la lista circular.

manora bogara on la mota circalar.	0-0	- \	

43. Reescriba la función Queue::remove para que elimine de \bigcirc \mathbf{r} \bigcirc \mathbf{r}

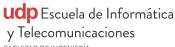
manera segura en la lista circular.	$\bigcirc I \bigcirc P \bigcirc C$





Instrucciones. Marque las casillas (○) completamente sin salirse de ellas (por ejemplo ●). Responda a los siguientes

(las respuestas en blanco no se t	oman en cuenta). Las últim	o una única opción. Se utilizar nas tres preguntas se evaluarán d e estas últimas tres preguntas, sin	le manera Completa (4 pts.),
$igcircle{0} igcircle{0} igcolumn{0} ig$	$\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 $	← Marque su RUT sin c cador (el número después de escriba sus nombres y apelli	el guión), y
$\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 4\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 5\bigcirc 6\bigcirc	$)4 \bigcirc 4 \bigcirc 4 \bigcirc 4$ $)5 \bigcirc 5 \bigcirc 5 \bigcirc 5$ $)6 \bigcirc 6 \bigcirc 6 \bigcirc 6$	Nombre(s) y apellido(s):	
$\begin{array}{c} \bigcirc 7 \bigcirc $	8 8 8 8 8		
1. ¿Qué es un sistema operativo? ———————————————————————————————————	O No es software, pero se	7. Tipo de kernel en el que la fun de él, y el resto se encuentra a niv	
que administra recursos y usuarios Es un software que admi-	encarga de administrar recursos y usuarios O Es un software que admi-	○ Microkernel○ Híbrido	
nistra usuarios 2. ¿Qué es una operación atómica Es una operación de	nistra recursos	8. ¿Qué es un pipe? O Es un buffer de usuario entre dos descriptores de	Es un buffer del kernel entre dos descriptores de
hardware que no puede ser dividida Es una operación indivi-	nel que no puede ser di- vidida Es una operación que	archivos Es un buffer de usuario entre dos procesos	archivos Es un buffer del kernel entre dos procesos
sible que no puede ser di- vidida	deshabilita las interrupciones	9. Región de la memoria reservad mantener el estado de las variable	= = = = = = = = = = = = = = = = = = = =
3. ¿Cuál de las siguientes no es criptor de archivo de UNIX?	una característica de un des-	funciones. Heap	O Swap
Se abren antes de usarSe cierran completamen-	Se leen a través de un bufferSe escriben a través de	Stack10. ¿Qué es la exclusión mutua?	○ RAM
te después de usar 4. ¿Cuál de las siguientes no es un	bytes	O Propiedad de la concurrencia para mantener	O Propiedad donde los hilos se bloquean mutua-
	Tiene un stack propioTiene un contador de programa propio	los hilos excluidos O Propiedad en la que solamente un hilo puede accesar simultáneamente	mente Objeto de sincronización que bloquea los hilos
5. Modo de ejecución del procesado	r donde no se ejecuta ninguna	11. Modo de ejecución del proc instrucción antes de ejecutarla.	esador en el que verifica cada
verificación. Modo de usuario Modo dual	○ Modo inseguro○ Modo kernel	○ Modo de usuario○ Modo dual	
6. ¿Cuál de las siguientes afirmaci		12. Es una señal síncrona al produn evento que requiere su atención	1 1
spin wait? Oroduce deadlocks	Es justificable en mono-	Señal de softwareSeñal de I/O	TrampaInterrupción
Es justificable en multi- procesadores	procesadores No se debe de usar nunca	13. ¿Cuál de los siguientes elementos?	entos no es compartido por los
		Variables globalesCódigo	○ Heap○ Stack



udp Escuela de Informática
y Telecomunicaciones
FACULTAD DE INGENIERÍA

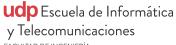
 14. ¿Qué son las direcciones de m Memoria que reserva el sistema operativo para el uso de máquinas virtuales Capa adicional que utiliza el sistema operativo en modo kernel 	emoria virtuales? Capa de indirección que le da flexibilidad al sistema operativo para administrar la memoria Memoria adicional que crea el sistema operativo para engañar a las aplicaciones	 23. ¿Qué es concurrencia? Realizar múltiples actividades una después de la otra Realizar una actividad sin la interrupción de otra 24. En la creación del proceso del circulato de correctora. 	
15. ¿Qué característica del sistema independientemente del hardware o Virtualización	en la computadora?	el siguiente después de cargar prog Informar al calendariza- dor que el programa se puede ejecutar	Inicializar el PCB en el kernel Copiar los argumentos en la memoria
Compartimiento de recursos	() Ilusionista	O Inicializar el hardware para que se ejecute el in- cio del programa	en la memoria
16. ¿Cuál de los siguientes eventos modo kernel a usuario?	s no genera una transición de	25. ¿Qué es el multi-hilado preve	ntivo?
Ocontinuar después de una interrupción	Creación de nuevo proceso	Es cuando los hilos que se ejecutan no pueden	Es cuando los hilos no usan el procesador hasta
Chamada I/O a un dispositivo	Cambiar a un proceso di- ferente	descalendarizarse Es cuando los hilos en-	que se les entrega indefi- nidamente
17. ¿Cuál de las siguientes es una	característica de fork?	tregan el procesador vo- luntariamente, no son in-	Es cuando los hilos que se ejecutan pueden ser
O Copiar a la memoria el	O Crear una copia del pro-	terrumpidos	cambiados indistinta-
código del programa del proceso padre	ceso padre, pero no pue- de ser confiado igual que		mente
O Copiar el proceso padre	él O C : 1	26. ¿Qué es una variable de cond Variable de sincroniza-	ición? Varialbe de sincroni-
completamente	Ocopiar el proceso padre con privilegios distintos	ción que permite la ex- clusión mutua	zación que implementa yield
18. ¿Donde almacena el sistema op	perativo la toda la información	Variable de sincroniza-	Objeto de sincronización
obre un proceso en particular? Process control block	O Memoria	ción que permite la espe- ra eficiente de un hilo	que implementa yield en modo usuario
(PCB) O Stack	Неар	27. ¿Cuántos procesos son creado guiente programa?	os durante la ejecución del si-
19. ¿Cuál es el estado de un hilo e	en el que está listo para ejecu-	garente programa.	
tarse pero no está en el procesador Listo (ready)		<pre>main(int argc, c forkthem(5);</pre>	char** argv){
Ejecutandose (running)	Inicializado (init)	<pre>} void forkthem(in if (n > 0){</pre>	nt n){
20. ¿Qué hace la función wait de		fork(); forkthem(n-1):
O Duerme el hilo atómica- mente mientras despier-	U Libera el seguro global, y duerme al hilo	}	.,
ta al siguiente	Atómicamente libera los	① 16	\bigcirc 4
 Mueve el hilo a la lista de espera atómicamente 	seguros y mueve el hilo a la lista de espera	$\bigcirc 30$	5
21. ¿Cuándo se da una condición O Varios hilos se ejecutan	de carrera? C Existe una sección críti-	28. ¿Cuál es el principal problema El bloqueo del proceso,	O No podemos tener varios
simultáneamente Varios hilos se pueden intercalar en su ejecución	ca O El estado de la ejecución de un programa depen-	bloquea todos los hilos No podemos calendari- zar hilos dentro del pro-	procesos por hilo No podemos tener varios hilos por proceso
simultánea	de del intercalado de di- ferentes hilos	ceso	
22. Llamada del sistema de UNIX	X que le permite a las aplica-	29. ¿Qué es una llamada de sister Funciones del kernel que	Ocódigo del kernel que
ciones el comunicarse entre sí para resumirlas.	a terminarias, suspenderias, o	permiten al usuario acce- sar a recursos restringi-	ejecuta código de usuario Funciones de hardware
O fork	O wait	dos	que llama el usuario
signal	exec	 Llamada que hace el ker- nel para realizar una ins- trucción 	

Para revisión, preferentemente imprimir documentos compilados desde automultiple-choice.



dientemente

30. Tipo de kernel en el que la réste se encuentra dentro de él.	nayoría de la funcionalidad de	37. Es una señal asíncrona al pro un evento que requiere su atención	
○ Híbrido○ Monolítico	○ Macrokernel○ Microkernel	☐ Interrupción☐ Señal de I/O	○ Señal de software○ Trampa
31. ¿Cuál de los siguientes eleme	ntos no pertenece al <i>TCB</i> ? O Puntero al stack frame	38. ¿Qué es la ejecución de una a gidos?	plicación con permisos restrin-
Estado del hilo	Registros salvados	○ Programa○ Hilo	O Proceso O Aplicación
 32. Región de la memoria reser para alojar estructuras de datos que su /li>	ue el proceso pueda necesitar. Heap Stack	 39. En la creación del proceso de el siguiente después de inicializar dor que el programa se puede ejecutar Copiar los argumentos en la memoria 40. ¿Cuál es el primer paso en la grama prog? Crear e inicializar el PCB en el kernel Inicializar la memoria 	el espacio de memoria? Cargar prog en la memoria Inicializar el hardware para que se ejecute el incio del programa
estado a esperando ($wai-ting$) Dado el siguiente código			ria
<pre>main(int argc, int child == : int x = 5; if (child == 0; x += 5; } else { child = for) x += 10; if (child) x+=5; } }</pre>	fork(); ();		
34. ¿Cuál de los siguientes no es x al término de los procesos del p			
○ 15○ 5	○ 20 ○ 10		
35. ¿Cuántas copias distintas de cución del programa anterior?	la variable ${\bf x}$ existen en la eje-		
\bigcirc 4 \bigcirc 2	$ \bigcirc \ 1 \\ \bigcirc \ 3$		
36. ¿Qué es un hilo? Una secuencia de ejecución que puede ser calendarizada independientemente Varias secuencias de ejecución que pueden ser calendarizadas indepen-	 Varias secuencias de ejecución que pueden ser calendarizadas dependientemente Una secuencia de ejecución que es calendarizada dependientemente 		



FACULTAD DE INGENIERÍA

41. Dado el siguiente código del programa main

```
1
   char bar(){
2
     int b = 2;
3
     return 'a';
4 }
   int foo(int a, char b){
6
     int i=0;
7
     bar();
   }
8
9
  main(){}
    int b=2;
11
      foo();
12
```

Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

 $\bigcirc I \bigcirc P \bigcirc C$ a la dirección de cada instrucción.

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```

Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de $\bigcirc \mathbf{r} \bigcirc \mathbf{p} \bigcirc \mathbf{c}$

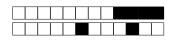
manera segura en la lista circular.	$\bigcirc I \bigcirc P \bigcirc C$

43. Reescriba la función Queue::remove para que elimine de manera segura en la lista circular $\bigcirc I \bigcirc D \bigcirc C$

nanera segura en la lista circular.	OI OP	\bigcirc C

Para revisión, preferentemente imprimir documentos compilados desde automultiple-choice.





cuestionamientos en las hojas qua (las respuestas en blanco no se	ue se le entregan marcando toman en cuenta). <i>Las últim</i>	salirse de ellas (por ejemplo o una única opción. Se utilizar as tres preguntas se evaluarán d estas últimas tres preguntas, sin	rá factor de corrección 4 a 1 e manera Completa (4 pts.),
$\begin{array}{c} \bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\bigcirc 0\\ \bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1\bigcirc 1\\ \bigcirc 2\bigcirc 2\bigcirc 2\bigcirc 2\bigcirc 2\\ \bigcirc 3\bigcirc 3\bigcirc 3\bigcirc 3\bigcirc 3 \bigcirc 3 \end{array}$	$\begin{array}{c} \begin{array}{c} \begin{array}{c} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \begin{array}{c} \end{array} \\ \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \end{array} \\ \begin{array}{c} \end{array} \\ \\ \end{array} \\ \\ \end{array} \\ \begin{array}{c} \end{array} \\ \\ \end{array} \\ \\ \end{array} \\ \begin{array}{c} \end{array} \\ \\ \end{array} \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \end{array} \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \\ \end{array} \\ \\ \end{array} \\ \begin{array}{c} \\ \\ \\ \end{array} \\ \\ \end{array} \\ \\ \\ \end{array} \\ \\ \\ \end{array} \\ \\ \end{array} \\ \\ \\ \\ \end{array} \\ \\ \\ \\ \end{array} \\$	← Marque su RUT sin c cador (el número después de escriba sus nombres y apelli	el guión), y
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	Nombre(s) y apellido(s):	
99999999 1. ¿Donde almacena el sistema op sobre un proceso en particular?	$\bigcirc 9 \bigcirc 9 \bigcirc 9 \bigcirc 9$	6. ¿Qué es una variable de condid O Variable de sincroniza-	ción? Variable de sincroniza-
Heap Memoria		ción que permite la espera eficiente de un hilo Objeto de sincronización que implementa yield	variable de sincronización que permite la exclusión mutua Varialbe de sincronización que implementa
2. Tipo de kernel en el que la m éste se encuentra dentro de él. Macrokernel Microkernel	ayoría de la funcionalidad de Monolítico Híbrido	en modo usuario 7. ¿Cuál es el principal problema O No podemos calendarizar hilos dentro del pro-	yield de los hilos a nivel de usuario? El bloqueo del proceso, bloquea todos los hilos
3. ¿Qué es una llamada de sistem. Clamada que hace el kernel para realizar una instrucción	a? Código del kernel que ejecuta código de usuario Funciones de hardware	ceso No podemos tener varios procesos por hilo 8. ¿Cuál de las siguientes afirmados.	No podemos tener varios hilos por proceso
Funciones del kernel que permiten al usuario acce- sar a recursos restringi- dos	que llama el usuario	spin wait? Es justificable en multiprocesadores No se debe de usar nunca	Produce deadlocksEs justificable en monoprocesadores
 4. ¿Qué es un hilo? Una secuencia de ejecución que es calendarizada dependientemente Varias secuencias de ejecución que pueden ser calendarizadas independientemente 	 Una secuencia de ejecución que puede ser calendarizada independientemente Varias secuencias de ejecución que pueden ser calendarizadas dependientemente 	 Qué es el multi-hilado preven Es cuando los hilos entregan el procesador voluntariamente, no son interrumpidos Es cuando los hilos no usan el procesador hasta que se les entrega indefi- 	tivo? Es cuando los hilos que se ejecutan pueden ser cambiados indistintamente Es cuando los hilos que se ejecutan no pueden descalendarizarse
5. Modo de ejecución del procesinstrucción antes de ejecutarla.	sador en el que verifica cada	nidamente 10. Región de la memoria reserpara alojar estructuras de datos q	
Modo de usuarioModo seguro	○ Modo dual○ Modo kernel	Swap RAM	Stack Heap
		11. ¿Cuál de los siguientes eleme Variables locales Puntero al stack frame	ntos no pertenece al <i>TCB</i> ? Estado del hilo Registros salvados



12. ¿Cuál de las siguientes es una car Copiar el proceso padre	Copiar el proceso padre	20. ¿Qué hace la función wait de la Libera el seguro global, y	O Duerme el hilo atómica-
con privilegios distintos Copiar a la memoria el código del programa del proceso padre	completamente Crear una copia del proceso padre, pero no puede ser confiado igual que	duerme al hilo Mueve el hilo a la lista de espera atómicamente	mente mientras despierta al siguiente Atómicamente libera los seguros y mueve el hilo a la lista de espera
13. ¿Qué característica del sistema op independientemente del hardware en la Portabilidad Virtualización 14. ¿Cuál de las siguientes no es una	a computadora? Compartimiento de recursos Ilusionista	 21. En la creación del proceso del el siguiente después de cargar prog O Informar al calendarizador que el programa se puede ejecutar O Inicializar el PCB en el kernel 	
Tiene un stack propio Tiene un contador de programa propio	programa Tiene un segmento de datos propio	22. ¿Qué es un pipe? Es un buffer del kernel entre dos procesos Es un buffer de usuario	Es un buffer del kernel entre dos descriptores de archivos
 ¿Qué es concurrencia? Realizar múltiples actividades una después de la otra Realizar múltiples actividades simultáneamente ¿Cuántos procesos son creados de la concessión de l	Realizar una actividad sin la interrupción de otra Realizar múltiples activi- dades sin la interrupción de otras	entre dos descriptores de archivos 23. En la creación del proceso del el siguiente después de inicializar el Copiar los argumentos en la memoria Inicializar el hardware	espacio de memoria? Cargar prog en la memoria Informar al calendariza-
<pre>guiente programa? main(int argc, char</pre>	** argv){	para que se ejecute el incio del programa 24. ¿Cuál de los siguientes eventos modo kernel a usuario? Continuar después de una interrupción Llamada I/O a un dispositivo	dor que el programa se puede ejecutar no genera una transición de Cambiar a un proceso diferente Creación de nuevo proceso
forkthem(n-1); } 0 16 0 5	4 30	25. Región de la memoria reservada mantener el estado de las variables funciones. Stack	por el sistema operativo para
17. Es una señal síncrona al procesac un evento que requiere su atención. O Señal de I/O	lor que indica que ocurrió Interrupción	Swap26. ¿Qué es la ejecución de una aper	Неар
Señal de 1/0 Señal de software 18. ¿Qué es la exclusión mutua?	Trampa	gidos? Programa Hilo	AplicaciónProceso
Objeto de sincronización que bloquea los hilos Propiedad de la concurrencia para mantener los hilos excluidos	Propiedad en la que solamente un hilo puede accesar simultáneamente Propiedad donde los hilos se bloquean mutuamente	Dado el siguiente código main(int argc, ch int child == fc int x = 5; if (child == 0) x += 5;	ork();
19. ¿Qué es una operación atómica? Es una operación de hardware que no puede ser dividida Es una operación indivisible que no puede ser dividida	Es una operación del kernel que no puede ser dividida Es una operación que deshabilita las interrupciones	<pre>x +- 5; } else { child = fork(x += 10; if (child) { x+=5; } }</pre>	();



27. ¿Cuántas copias distintas de cución del programa anterior? 3 1	la variable \mathbf{x} existen en la eje- $ \bigcirc \ 4 $ $ \bigcirc \ 2 $	36. ¿Cuál de las siguientes no es criptor de archivo de UNIX? Se cierran completamente después de usar	O Se abren antes de usar O Se leen a través de u
28. ¿Cuál de los siguientes no es x al término de los procesos del pr		Se escriben a través de bytes37. Es una señal asíncrona al pro	
O 20	O 15	un evento que requiere su atención Trampa Señal de software	on. O Interrupción O Señal de I/O
 29. Tipo de kernel en el que la fun de él, y el resto se encuentra a nive	el de usuario. Microkernel Monolítico	 38. ¿Cuándo se da una condición El estado de la ejecución de un programa depende del intercalado de diferentes hilos Varios hilos se pueden intercalar en su ejecución simultánea 39. ¿Cuál de los siguientes elementes 	n de carrera? Varios hilos se ejecuta simultáneamente Existe una sección crít ca
31. Modo de ejecución del proces guna verificación. Modo de usuario Modo kernel	·	hilos? Heap Stack 40. ¿Qué hace la función yield d	
32. ¿Cuál es el estado de un hilo o tarse pero no está en el procesador O Inicializado (init) O Listo (ready)		 Duerme al hilo por un tiempo determinado, y cambia el estado a esperando (waiting) Entrega voluntariamente 	 Entrega voluntariament el resto del ciclo en procesador, y cambia e estado a listo (ready) Entrega voluntariament
33. Llamada del sistema de UNI ciones el comunicarse entre sí para resumirlas.	X que le permite a las aplica-	el resto del ciclo en el procesador, y cambia el estado a terminado (finished)	el resto del ciclo en e procesador, y cambia e estado a esperando (wa ting)
34. ¿Cuál es el primer paso en la grama prog? Inicializar la memoria Copiar prog a la memoria	Inicializar el hardware para que se ejecute el proceso desde el inicio Crear e inicializar el PCB en el kernel		
 35. ¿Qué son las direcciones de m Memoria que reserva el sistema operativo para el uso de máquinas virtuales Memoria adicional que crea el sistema operativo para engañar a las aplicaciones 	nemoria virtuales? Capa adicional que utiliza el sistema operativo en modo kernel Capa de indirección que le da flexibilidad al sistema operativo para administrar la memoria		



```
1
   char bar(){
2
     int b = 2;
3
     return 'a';
4 }
   int foo(int a, char b){
6
     int i=0;
7
      bar();
   }
8
9
   main(){}
     int b=2;
11
      foo();
12
```

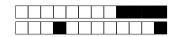
Dibuje el stack completo del programa cuando se encuentra ejecutando la linea número 2. Utilice el número de línea para referirse

a la dirección de cada instrucción.

OI OP OC

El siguiente código implementa una lista circular.

```
const int MAX = 10;
class Queue{
  int items[MAX];
  int front;
  int nextEmpty;
public:
  Queue() { front = nextEmpty = 0;}
  ~Queue();
  void insert(int item);
  int remove();
void Queue::insert(int item){
  items[nextEmpty%MAX] = item;
  nextEmpty++;
int Queue::remove(){
  int item;
  item = items[front%MAX];
  front++;
  return item;
```



Sin embargo, las funciones Queue::insert y Queue::remove no están sincronizadas. Por lo que más elementos de los que soporta la lista pueden insertarse o removerse.

Responda lo que se le solicita a continuación. Puede declarar los Lock y CV que considere necesarios. Tenga en cuenta que la implementación debe de considerar que no puede insertar si la lista está llena, por lo que deberá de detener la función que está insertando. De manera similar, cuando elimine si no hay ningún elemento en la lista deberá esperar a que uno exista.

42. Reescriba la función Queue::insert para que inserte de manera segura en la lista circular

manera segura en la fista circular.	\bigcirc

43. Reescriba la función Queue::remove para que elimine de manera segura en la lista circular.

manera segura en la lista circular.	OI OP OC