

<b>Comenzado el</b>	martes, 27 de abril de 2021, 21:00
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	martes, 27 de abril de 2021, 21:40
<b>Tiempo empleado</b>	39 minutos 26 segundos
<b>Calificación</b>	<b>1,500</b> de 10,000 ( <b>15%</b> )

Pregunta **1**

Correcta

Puntúa 0,625 sobre 0,625

Procedimiento Call Gate

Se proporciona un puntero lejano al Gate como operando de destino en una instrucción CALL o JMP .

El selector de segmento de este puntero identifica la Call Gate.

Se utiliza el selector de Call Gate para localizar el descriptor de segmento para el segmento de código de destino.

Luego combina la dirección base del descriptor de segmento de código con el offset del puntero lejano para formar la dirección lineal del punto de entrada del procedimiento en el segmento de código.

Seleccione una:

- ☒ Verdadero ✖
- ☐ Falso

La respuesta correcta es 'Falso'

Pregunta **2**

Incorrecta

Puntúa 0,000 sobre 0,625

¿Qué es un proceso?

Seleccione una:

- ☐ a. Un proceso es un programa listo para ser ejecutado.
- ☐ b. Ninguna de las respuestas es válida.
- ☒ c. Un proceso es una instancia de un programa en ejecución que solo se puede ejecutar en espacio de usuario. ✗
- ☐ d. Un proceso es el número que identifica a un programa en ejecución.
- ☐ e. Un proceso es una instancia de un programa en ejecución mas el estado del mismo.
- ☐ f. Un proceso es una instancia de un programa en ejecución que solo puede ejecutar invocando el sistema operativo.

Respuesta incorrecta.

La respuesta correcta es: Un proceso es una instancia de un programa en ejecución mas el estado del mismo.

Pregunta **3**

Parcialmente correcta

Puntúa 0,250 sobre 1,250

Dos procesos relacionados se comunican por medio de una FIFO.

- 1- El proceso hijo abre la FIFO con permisos de solo escritura y escribe "0123456789" en la FIFO
- 2- El proceso hijo cierra y elimina la FIFO y luego termina él.
- 3- El proceso padre abre la FIFO con permisos de solo lectura, lee la FIFO y muestra lo leído.
- 4- El proceso padre espera a que el proceso hijo termine para cerrar la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define BB "/tmp/BB"
int b[5],c,e;
char d[10];
int main(){
    mkfifo(BB, 0777);
    b[0]=fork();
    a = open(BB, O_RDWR, 0); ✖
    if(b[0]==0){
        close(c); ✖
        write(c, E,sizeof(E));
        c = open(BB, O_WRONLY, 0); ✖
        read(c, E,sizeof(E)); ✖
        exit(0);
    } else{
        ✖
        e = read(c, d, sizeof(d));
        write(STDOUT_FILENO, d, e);
        wait(NULL);
        close(c);
    }
    exit(0);
}
c = open(BB, O_RDWR, 0); write(a, d, sizeof(d)); close(e); write(c, E,sizeof(E)); c = open(BB, O_RDONLY, 0); unlink(BB); //línea en blanco
.
```

Respuesta parcialmente correcta.

Ha seleccionado correctamente 0.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una FIFO.

- 1- El proceso hijo abre la FIFO con permisos de solo escritura y escribe "0123456789" en la FIFO
- 2- El proceso hijo cierra y elimina la FIFO y luego termina él.
- 3- El proceso padre abre la FIFO con permisos de solo lectura, lee la FIFO y muestra lo leído.
- 4- El proceso padre espera a que el proceso hijo termine para cerrar la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define BB "/tmp/BB"
int b[5],c,e;
char d[10];
int main(){
    mkfifo(BB, 0777);
```

```
b[0]=fork();
[//linea en blanco]
if(b[0]==0){
    [c = open(BB, O_WRONLY, 0);]
    write(c, E, sizeof(E));
    [close(c);]
    [unlink(BB);]
    exit(0);
} else{
    [c = open(BB, O_RDONLY, 0);]
    e = read(c, d, sizeof(d));
    write(STDOUT_FILENO, d, e);
    wait(NULL);
    close(c);
}
exit(0);
}
```

#### Pregunta 4

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmeLFJNcP>

2) Modifique el archivo padre.c para que envíe una señal SIGUSR1 al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGUSR1
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 652

Pregunta **5**

Incorrecta

Puntúa 0,000 sobre 0,625

El campo Back Link del TSS de la tarea actual, contiene el selector de segmento de TSS de la tarea anterior. Este campo permite que se inicie un cambio de tarea a la tarea anterior cuando se ejecuta una instrucción JMP, CALL o IRET.

Seleccione una:

- ☒ Verdadero **✗**
- ☐ Falso

La respuesta correcta es 'Falso'

Pregunta **6**

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Para la técnica IPC pipe, una lo que corresponda:

Si un proceso intenta escribir en el extremo de escritura de una tubería que está llena

el proceso se bloquea. ▾



Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía y todos sus descriptores de escritura fueron cerrados

el proceso termina. ▾



Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía

da error de lectura y termina. ▾



Si un proceso intenta escribir en el extremo de escritura de una tubería que está vacía

el proceso logra escribir. ▾



Si un proceso intenta leer en el extremo de lectura de una tubería que está llena

el proceso lee datos. ▾



Si un proceso intenta escribir en el extremo de escritura de una tubería que cerró todos los descriptores de lectura

el proceso lee 0 bytes. ▾



Respuesta parcialmente correcta.

Ha seleccionado correctamente 3.

La respuesta correcta es: Si un proceso intenta escribir en el extremo de escritura de una tubería que está llena → el proceso se bloquea., Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía y todos sus descriptores de escritura fueron cerrados → el proceso lee 0 bytes., Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía → el proceso se bloquea., Si un proceso intenta escribir en el extremo de escritura de una tubería que está vacía → el proceso logra escribir., Si un proceso intenta leer en el extremo de lectura de una tubería que está llena → el proceso lee datos., Si un proceso intenta escribir en el extremo de escritura de una tubería que cerró todos los descriptores de lectura → el proceso termina.

Pregunta **7**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/wygC5eEcTrAR56t>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la tubería declarada en la variable fd.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la tubería, imprimirá por consola dos líneas:

```
Leido desde tubería: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 537

Pregunta **8**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione lo correcto para tipos de estructura de sistemas operativos

- ☒ a. Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers. ✓
- ☒ b. Los OS cliente/servidor se caracterizan por separar multiprogramación de máquina extendida. ✗
- ☐ c. Los OS cliente/servidor son una generalización de OS microkernel .
- ☐ d. Hasta los OS monolíticos pueden tener una estructura interna.
- ☐ e. Los OS cliente/servidor se ejecutan mas rápido que los OS monolíticos.
- ☒ f. Los OS Máquina Virtual se basan en organizar las funciones del OS de manera jerárquica. ✗

Respuesta incorrecta.

Las respuestas correctas son: Hasta los OS monolíticos pueden tener una estructura interna., Los OS cliente/servidor son una generalización de OS microkernel ., Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers.

Pregunta 9

Incorrecta

Puntúa 0,000 sobre 0,625

Para una arquitectura IA32, indique cuál de las siguientes definiciones es **falsa**:

Seleccione una:

- ☒ a. Un acierto de TLB permite ahorrar dos accesos a memoria física. ✖
- ☐ b. La parte oculta de los registros de segmento permite ahorrar un acceso a memoria (mientras no se cambie de segmento).
- ☐ c. La TLB es una caché de traslación de direcciones lineales a físicas.
- ☐ d. Esta opción fue eliminada después de que el intento fuera iniciado.
- ☐ e. En modo protegido, es obligatorio al menos el uso de paginación para trasladar direcciones.

Respuesta incorrecta.

La respuesta correcta es: En modo protegido, es obligatorio al menos el uso de paginación para trasladar direcciones.

Pregunta **10**

Parcialmente correcta

Puntúa 0,313 sobre 1,250

En el siguiente programa se ejecutan 4 procesos.

1. El proceso padre (padre1) crea un hijo (hijo1).
2. El proceso padre y el hijo1 crean otros hijos (hijo2 e hijo3).
3. Los procesos padres deben enviar la señal SIGKILL a los procesos hijos.
4. La línea **printf ("Mi pid es %d\n",getpid());** la debe ejecutar solo el proceso padre (padre1).

Completar:

```
int a,b=0;
int main (){
    a = fork();
    b = fork();
    // linea en blanco ✓
    if (a==0) {
        if(b==0) {
            c = fork(); ✗
            while(1);
        }else{
            wait(NULL); ✗
            while(1);
        }
    }
    if (a>0) {
        if(b==0) {
            while(1);
        }
    }
    sleep(2);
    kill(b, SIGKILL); ✗
    kill(b, SIGKILL);
    printf ("Mi pid es %d\n",getpid());
    exit(0);
}
```

b = fork();

signal(SIGKILL,a);

kill(c, SIGKILL);

kill(a, SIGKILL);

signal(SIGKILL,b);

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es:

En el siguiente programa se ejecutan 4 procesos.

1. El proceso padre (padre1) crea un hijo (hijo1).
2. El proceso padre y el hijo1 crean otros hijos (hijo2 e hijo3).
3. Los procesos padres deben enviar la señal SIGKILL a los procesos hijos.
4. La línea **printf ("Mi pid es %d\n",getpid());** la debe ejecutar solo el proceso padre (padre1).

Completar:



```

int a,b=0;
int main (){
    a = fork();
    b = fork();
    [// linea en blanco]
    if (a==0) {
        if(b==0) {
            [// linea en blanco]
            while(1);
        }else{
            [kill(b, SIGKILL);]
            while(1);
        }
    }
    if (a>0) {
        if(b==0) {
            while(1);
        }
    }
    sleep(2);
    [kill(a, SIGKILL);]
    kill(b, SIGKILL);
    printf ("Mi pid es %d\n",getpid());
    exit(0);
}

```

Pregunta **11**

Incorrecta

Puntúa 0,000 sobre 0,625

Indique todas las afirmaciones verdaderas en la traslación de direcciones lógicas a lineales para Intel IA-32:

Seleccione una o más de una:

- ☐ a. El registro IP se utiliza de offset cuando en el selector se pone el registro DS.
- ☒ b. Los bits de BASE de un descriptor de segmento indican una dirección lineal de 32 bits desde donde inicia el segmento. ✓
- ☒ c. El SP se utiliza de offset cuando en el selector se pone el registro SS. ✓
- ☐ d. Se debe desplazar a la izquierda 3 bits los 13 bits de mas peso del LDTR para acceder a una entrada de la LDT.
- ☐ e. Se debe desplazar a la izquierda 3 bits el índice del selector para acceder a una entrada de la GDT.
- ☐ f. Es obligatorio que la GDT contenga al menos un descriptor de LDT.
- ☒ g. El LDTR indica la dirección base de la LDT. ✗

Respuesta incorrecta.

Las respuestas correctas son: Los bits de BASE de un descriptor de segmento indican una dirección lineal de 32 bits desde donde inicia el segmento., El SP se utiliza de offset cuando en el selector se pone el registro SS., Se debe desplazar a la izquierda 3 bits el índice del selector para acceder a una entrada de la GDT.

Indique la afirmación correcta sobre la syscall signal():

Seleccione una:

- ☐ a. Se utiliza para enviar una señal específica a un proceso.
- ☐ b. Se utiliza para ignorar la recepción de la señal SIGKILL.
- ☐ c. Se utiliza para terminar a ejecución de un proceso.
- ☐ d. Se utiliza para ignorar algunas señales determinadas.
- ☐ e. Se utiliza para terminar un proceso del cual se conoce el PID.
- ☐ f. Se utiliza para poder ignorar todo tipo de señales.
- ☐ g. Ninguna respuesta es válida.

Respuesta incorrecta.

La respuesta correcta es: Se utiliza para ignorar algunas señales determinadas.

◀ Avisos

Ir a...



<b>Comenzado el</b>	martes, 27 de abril de 2021, 21:00
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	martes, 27 de abril de 2021, 21:29
<b>Tiempo empleado</b>	29 minutos 4 segundos
<b>Calificación</b>	<b>4,646</b> de 10,000 ( <b>46%</b> )

Pregunta **1**

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Una lo correcto para multitasking en IA32:

El DPL del TSS descriptor tiene	el selector de segmento para el LDT de la tarea.	⌵	✗
El back link del TSS tiene	el selector de segmento para el TSS de la tarea anterior.	⌵	✓
El TR tiene	el selector del TSS de la tarea que se está ejecutando actualmente.	⌵	✓
En el TSS tiene	el estado de la tarea está activa	⌵	✗

Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

La respuesta correcta es: El DPL del TSS descriptor tiene → la misma función que el de un descriptor de segmento de datos., El back link del TSS tiene → el selector de segmento para el TSS de la tarea anterior., El TR tiene → el selector del TSS de la tarea que se está ejecutando actualmente., En el TSS tiene → el selector de segmento para el LDT de la tarea.

Pregunta **2**

Correcta

Puntúa 0,625 sobre 0,625

Para una arquitectura IA32, indique cuál de las siguientes definiciones es **falsa**:

Seleccione una:

- ☐ a. Un acierto de TLB permite ahorrar dos accesos a memoria física.
- ☒ b. La TLB es una caché de traslación de direcciones lógicas a físicas. ✓
- ☐ c. En modo protegido, es obligatorio al menos el uso de segmentación para trasladar direcciones.
- ☐ d. Un acierto de TLB permite ahorrar dos accesos a direcciones lógicas.
- ☐ e. La parte oculta de los registros de segmento permite ahorrar un acceso a memoria (mientras no se cambie de segmento).

Respuesta correcta

La respuesta correcta es: La TLB es una caché de traslación de direcciones lógicas a físicas.

Pregunta **3**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione lo correcto para tipos de estructura de sistemas operativos

- ☐ a. Los OS cliente/servidor son una generalización de OS microkernel .
- ☒ b. Los OS cliente/servidor se caracterizan por separar multiprogramación de máquina extendida. ✗
- ☐ c. Los OS cliente/servidor se ejecutan mas rápido que los OS monolíticos.
- ☐ d. Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers.
- ☒ e. Los OS Máquina Virtual se basan en organizar las funciones del OS de manera jerárquica. ✗
- ☒ f. Hasta los OS monolíticos pueden tener una estructura interna. ✓

Respuesta incorrecta.

Las respuestas correctas son: Hasta los OS monolíticos pueden tener una estructura interna., Los OS cliente/servidor son una generalización de OS microkernel ., Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers.

Pregunta **4**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsl>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/yoPHJsGT35eDPfW>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la tubería declarada en la variable fd.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la tubería, imprimirá por consola dos líneas:

```
Leído desde tubería: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 258

Pregunta **5**

Incorrecta

Puntúa 0,000 sobre 0,625

Indique que información se guarda en cada entrada de la tabla de procesos (PCB):

Seleccione una o más de una:

- ☐ a. ID de los hilos que está usando.
- ☒ b. IPC que usa actualmente el proceso.
- ☒ c. PID del hijo.
- ☐ d. El valor del contador de programa.
- ☐ e. Descriptores de archivos abiertos.

✗

✗

Respuesta incorrecta.

Las respuestas correctas son: El valor del contador de programa., Descriptores de archivos abiertos.

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo lee de la FIFO, muestra lo leído, cierra la FIFO y luego termina él.
2. El proceso padre escribe "0123456789" en la FIFO, espera a que el proceso hijo termine.
3. El proceso padre para cierra y elimina la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define CC "/tmp/BB"
int e,b,f,g;
char d[10];
int main(){
    mkfifo(CC, 0777);
    b=fork();
    if(b>0){
         ✓
        f = open(CC, O_WRONLY, 0);
         ✓
        wait(NULL);
        close(f);
         ✗
        exit(0);
    }
     ✓
    e = open(CC, O_RDONLY, 0);
     ✗
    write(STDOUT_FILENO, d, g);
     ✓
    exit(0);
}
```

close(f)

read(f, E,sizeof(E));

exit(0);

pipe(e);

write(f, E,sizeof(E));

f = open(CC, O\_RDWR, 0);

g = write(e, d, sizeof(d));

kill(SIGUSR1,x);

close(e);

read(e, d, sizeof(d));

unlink(CC);

//línea en blanco

g = read(e, d, sizeof(d));

Respuesta parcialmente correcta.

Ha seleccionado correctamente 4.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo lee de la FIFO, muestra lo leído, cierra la FIFO y luego termina él.
2. El proceso padre escribe "0123456789" en la FIFO, espera a que el proceso hijo termine.
3. El proceso padre para cierra y elimina la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define CC "/tmp/BB"
int e,b,f,g;
char d[10];
int main(){
    mkfifo(CC, 0777);
    b=fork();
    if(b>0){
        [//línea en blanco]
        f = open(CC, O_WRONLY, 0);
        [write(f, E,sizeof(E));]
        wait(NULL);
        close(f);
        [unlink(CC);]
        exit(0);
    }
    [//línea en blanco]
    e = open(CC, O_RDONLY, 0);
    [g = read(e, d, sizeof(d));]
    write(STDOUT_FILENO, d, g);
    [close(e);]
    exit(0);
}
```

## Pregunta 7

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/W5qgYjCgPLFxRnF>

2) Modifique el archivo padre.c para que envíe una señal SIGUSR2 al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGUSR2
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 244

Pregunta **8**

Parcialmente correcta

Puntúa 1,000 sobre 1,250

En el programa hay 3 procesos.

1. El proceso padre crea un hijo (hijo1) el cual queda en una espera activa.
2. Luego el proceso padre crea otro hijo (hijo2) el cual queda en una espera activa.
3. El proceso padre debe enviar la señal SIGKILL a sus dos hijos y ser el único proceso en ejecutar la línea **printf ("Mi pid es %d\n",getpid());**

Completar:

```
int a,b=0;
int main (){
    a= fork();
    [input] ✓
    if (a==0) {
        while(1);
        [input] ✓
        exit(0);
    }
    [input] ✓
    if (b==0) {
        while(1);
        [input] ✓
        exit(0);
    }
    [input] ✗
    kill(b,SIGKILL);
    printf ("Mi pid es %d\n", getpid());
    sleep(2);
    exit(0);
}
```

kill(a,SIGKILL);

signal(SIGKILL,a);

signal(SIGKILL,b);

kill(b,SIGKILL);

b=fork();

wait(NULL);

c=fork();

//línea en blanco

Respuesta parcialmente correcta.

Ha seleccionado correctamente 4.

La respuesta correcta es:

En el programa hay 3 procesos.

1. El proceso padre crea un hijo (hijo1) el cual queda en una espera activa.
2. Luego el proceso padre crea otro hijo (hijo2) el cual queda en una espera activa.
3. El proceso padre debe enviar la señal SIGKILL a sus dos hijos y ser el único proceso en ejecutar la línea **printf ("Mi pid es %d\n",getpid());**

Completar:



```
int a,b=0;
int main (){
    a= fork();
    [//línea en blanco]
    if (a==0) {
        while(1);
        [//línea en blanco]
        exit(0);
    }
    [b=fork();]
    if (b==0) {
        while(1);
        [//línea en blanco]
        exit(0);
    }
    [kill(a,SIGKILL);]
    kill(b,SIGKILL);
    printf ("Mi pid es %d\n", getpid());
    sleep(2);
    exit(0);
}
```

Pregunta **9**

Correcta

Puntúa 0,625 sobre 0,625

La arquitectura IA32 utiliza numerosos direccionamientos relativos (base / desplazamiento) para procesar una dirección lógica y finalmente obtener una dirección física. Indique el desplazamiento que corresponde a cada base.

Descriptor de segmento	Desplazamiento (dirección lógica).	⚙	✓
GDTR	Selector.	⚙	✓
CR3 (PDBR).	Dirección lineal (campo Directorio).	⚙	✓
PDE.	Dirección lineal (campo Tabla).	⚙	✓
PTE	Dirección lineal (campo Desplazamiento).	⚙	✓

Respuesta correcta

La respuesta correcta es: Descriptor de segmento → Desplazamiento (dirección lógica)., GDTR → Selector., CR3 (PDBR). → Dirección lineal (campo Directorio)., PDE. → Dirección lineal (campo Tabla)., PTE → Dirección lineal (campo Desplazamiento).

Pregunta **10**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione cuál **NO** es una motivación para utilizar IPC:

Seleccione una:

- ☐ a. Permitir la comunicación entre procesos relacionados.
- ☐ b. Compartir información entre procesos cooperativos.
- ☐ c. Ninguna respuesta es válida.
- ☐ d. Acelerar la ejecución de tareas, implementándolas en distintos procesos.
- ☐ e. Permitir la comunicación entre procesos.
- ☐ f. Permitir la modularidad.



Respuesta incorrecta.

La respuesta correcta es: Permitir la comunicación entre procesos.

Pregunta **11**

Correcta

Puntúa 0,625 sobre 0,625

Seleccione lo correcto respecto a señales

Seleccione una:

- ☐ a. Una señal es la petición por parte del usuario para la creación de un nuevo proceso.
- ☐ b. Un proceso puede bloquear la recepción de todas las señales.
- ☒ c. Una señal es una notificación entregada a un proceso debido a un evento asíncrono.
- ☐ d. Solo se pueden enviar señales entre procesos relacionados (padre-hijo-nieto).
- ☐ e. Las señales se generan por una interrupción debida a un evento externo.



Respuesta correcta

La respuesta correcta es: Una señal es una notificación entregada a un proceso debido a un evento asíncrono.

Pregunta **12**

Correcta

Puntúa 0,625 sobre 0,625

Para Protección por Segmentación, indique en que caso(s) no podrá hacer un acceso a datos de otro segmento de acuerdo a los valores de PL.

Seleccione una o más de una:

☒ a. CPL=3, DPL=2, RPL=1.



☒ b. RPL=3, DPL=2, CPL=0.



☐ c. RPL=2, CPL=1, DPL=2.

☐ d. DPL=3, CPL=1, RPL=2.

☐ e. DPL=2, RPL=1, CPL=1.

Respuesta correcta

Las respuestas correctas son: RPL=3, DPL=2, CPL=0., CPL=3, DPL=2, RPL=1.

[◀ Avisos](#)

Ir a...



**Comenzado el** martes, 27 de abril de 2021, 21:05

**Estado** Finalizado

**Finalizado en** martes, 27 de abril de 2021, 21:40

**Tiempo empleado** 34 minutos 10 segundos

**Calificación** 1,875 de 10,000 (19%)

Pregunta **1**

Parcialmente correcta

Puntúa 0,104 sobre 0,625

ordene los pasos en los que se lleva a cabo la llamada a sistema POSIX read()

- |   |   |   |
|---|---|---|
| 1 | ejecución de la call gate                                       | ✗ |
| 2 | llamada al procedimiento de la biblioteca                       | ✓ |
| 3 | ejecución de la task gate                                       | ✗ |
| 4 | retornar del programa de usuario al procedimiento de biblioteca | ✗ |
| 5 | llamada al procedimiento de la biblioteca                       | ✗ |
| 6 | ejecución del controlador de periférico de espacio kernel       | ✗ |

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es: 1 → hacer push de los parámetros de la syscall a la pila en orden inverso, 2 → llamada al procedimiento de la biblioteca, 3 → ejecución de la call gate, 4 → ejecución del controlador de periférico de espacio kernel, 5 → retornar al procedimiento de biblioteca de espacio usuario, 6 → retornar del procedimiento de biblioteca al programa de usuario

Pregunta **2**

Parcialmente correcta

Puntúa 0,521 sobre 0,625

Indique dónde se almacena cada estructura de datos en IA32:

El segmento de datos se almacena en	el espacio físico de direcciones.	✗
El Directorio de páginas se almacena en	el espacio físico de direcciones.	✓
La Tabla de páginas se almacena en	el espacio físico de direcciones.	✓
El GDTR se almacena en	el registros del procesador.	✓
La GDT se almacena en	el espacio lineal de direcciones.	✓
El segmento de código se almacena en	el espacio lineal de direcciones.	✓

Respuesta parcialmente correcta.

Ha seleccionado correctamente 5.

La respuesta correcta es: El segmento de datos se almacena en → el espacio lineal de direcciones., El Directorio de páginas se almacena en → el espacio físico de direcciones., La Tabla de páginas se almacena en → el espacio físico de direcciones., El GDTR se almacena en → el registros del procesador., La GDT se almacena en → el espacio lineal de direcciones., El segmento de código se almacena en → el espacio lineal de direcciones.

Pregunta **3**

Correcta

Puntúa 0,625 sobre 0,625

El campo Back Link del TSS de la tarea actual, contiene el selector de segmento de TSS de la tarea anterior. Este campo permite que se inicie un cambio de tarea a la tarea anterior cuando se ejecuta una instrucción IRET.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

Pregunta **4**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmeLFJNcP>

2) Modifique el archivo padre.c para que envíe una señal SIGUSR2 al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGUSR2
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 148

Pregunta **5**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione las afirmaciones correctas respecto a Posix Pipes:

Seleccione una o más de una:

- ☐ a. Se destruyen al cerrar los descriptores de lectura.
- ☐ b. Permiten sincronizar procesos relacionados.
- ☒ c. Se pueden abrir en forma bloqueantes y no bloqueantes.
- ☐ d. Se utilizan para enviar datos entre procesos no relacionados.
- ☐ e. Se destruyen al cerrar todos los descriptores de lectura y escritura.
- ☒ f. Poseen persistencia de kernel.
- ☐ g. Poseen un nombre en el sistema de archivos.
- ☒ h. Permiten el intercambio de datos entre procesos relacionados.



Respuesta incorrecta.

Las respuestas correctas son: Se destruyen al cerrar todos los descriptores de lectura y escritura., Permiten sincronizar procesos relacionados., Permiten el intercambio de datos entre procesos relacionados.

Pregunta **6**

Incorrecta

Puntúa 0,000 sobre 1,250

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo lee de la FIFO, muestra lo leído, cierra la FIFO y luego termina él.
2. El proceso padre escribe "0123456789" en la FIFO, espera a que el proceso hijo termine.
3. El proceso padre para cierra y elimina la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define CC "/tmp/BB"
int e,b,f,g;
char d[10];
int main(){
    mkfifo(CC, 0777);
    b=fork();
    if(b>0){
        write(f, E,sizeof(E));
        f = open(CC, O_WRONLY, 0);
        read(f, E,sizeof(E));
        wait(NULL);
        close(f);
        //linea en blanco
        exit(0);
    }
    e = open(CC, O_RDONLY, 0);
    write(STDOUT_FILENO, d, g);
    exit(0);
}
```

g = write(e, d, sizeof(d));

kill(SIGUSR1,x);

exit(0);

unlink(CC);

close(f)

read(e, d, sizeof(d));

pipe(e);

f = open(CC, O\_RDWR, 0);

g = read(e, d, sizeof(d));

close(e);

Respuesta incorrecta.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo lee de la FIFO, muestra lo leído, cierra la FIFO y luego termina él.
2. El proceso padre escribe "0123456789" en la FIFO, espera a que el proceso hijo termine.
3. El proceso padre para cierra y elimina la FIFO y luego terminar él.

Completar:

```

#define E "0123456789"
#define CC "/tmp/BB"
int e,b,f,g;
char d[10];
int main(){
    mkfifo(CC, 0777);
    b=fork();
    if(b>0){
        [//línea en blanco]
        f = open(CC, O_WRONLY, 0);
        [write(f, E,sizeof(E));]
        wait(NULL);
        close(f);
        [unlink(CC);]
        exit(0);
    }
    [//línea en blanco]
    e = open(CC, O_RDONLY, 0);
    [g = read(e, d, sizeof(d));]
    write(STDOUT_FILENO, d, g);
    [close(e);]
    exit(0);
}

```

Pregunta **7**

Correcta

Puntúa 0,625 sobre 0,625

Indique cuál de las siguientes transiciones de estados de un proceso NO es válida.

Seleccione una o más de una:

- ☐ a. Running a Blocked.
- ☒ b. Ready a Finished.
- ☐ c. New a Ready.
- ☐ d. Running a Ready.
- ☐ e. Ready a Running.
- ☐ f. Running a Finished.
- ☐ g. Blocked a Ready.
- ☒ h. Block a Running.



Respuesta correcta

Las respuestas correctas son: Ready a Finished., Block a Running.



Pregunta **8**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione lo correcto respecto a señales

Seleccione una:

- ☐ a. Una señal se usa generalmente para comunicar estados entre procesos.
- ☐ b. Un proceso puede bloquear la recepción de todas las señales.
- ☒ c. Se pueden enviar señales entre procesos no relacionados, a excepción de SIGSTOP.
- ☐ d. Una señal es la petición por parte del usuario para la terminación de un proceso.
- ☐ e. Las señales se generan exclusivamente con la syscall kill().



Respuesta incorrecta.

La respuesta correcta es: Una señal se usa generalmente para comunicar estados entre procesos.

Pregunta **9**

Sin contestar

Puntúa como 0,625

Indique todas las afirmaciones correctas para mecanismos de protección de IA32.

Seleccione una o más de una:

- ☐ a. La segmentación cuenta con protección por tipo de página.
- ☐ b. En la paginación, el nivel 0 de privilegios es el de mayor privilegio.
- ☐ c. La protección por privilegios previene a un programa acceder a un segmento de mayor privilegio.
- ☐ d. La paginación cuenta con protección por dominio direccionable.
- ☐ e. Las task gates permiten acceder a un nivel de privilegio mayor.
- ☐ f. La paginación cuenta con protección por chequeo de Límite.
- ☐ g. La paginación cuenta con chequeo de límite.

Respuesta incorrecta.

Las respuestas correctas son: La protección por privilegios previene a un programa acceder a un segmento de mayor privilegio., La paginación cuenta con protección por dominio direccionable.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d,e=0;
7
8 int main (){
9     a = fork();
10    b = fork();
11    printf ("1-Mi pid es %d\n", getpid());
12    if (a==0) { |
13        exit(0);
14    }
15    d = fork();
16    printf ("2-Mi pid es %d\n", getpid());
17    if (b==0) {
18        exit(0);
19    }
20    printf ("3-Mi pid es %d\n", getpid());
21    if (d>0) {
22        c = fork();
23        sleep(2);
24        exit(0);
25    }
26    printf ("4-Mi pid es %d\n", getpid());
27    exit(0);
28 }
29
```

Colocar la cantidad de procesos que ejecutan las líneas

11:

16:

20:

26:

7	4	3	2	0	6	5	1	8
---	---	---	---	---	---	---	---	---

Respuesta incorrecta.

La respuesta correcta es:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d,e=0;
7
8 int main (){
9     a = fork();
10    b = fork();
11    printf ("1-Mi pid es %d\n", getpid());
12    if (a==0) { |
13        exit(0);
14    }
15    d = fork();
16    printf ("2-Mi pid es %d\n", getpid());
17    if (b==0) {
18        exit(0);
19    }
20    printf ("3-Mi pid es %d\n", getpid());
21    if (d>0) {
22        c = fork();
23        sleep(2);
24        exit(0);
25    }
26    printf ("4-Mi pid es %d\n", getpid());
27    exit(0);
28 }
29

```

Colocar la cantidad de procesos que ejecutan las líneas

11: [4]

16: [4]

20: [2]

26: [1]

Pregunta **11**

Sin contestar

Puntúa como 0,625

Para una arquitectura IA32, se intenta acceder a memoria física y existe un acierto de TLB. Indique cuál afirmación es correcta (comparada con un fallo de TLB):

Seleccione una:

- ☐ a. Permite ahorrar un acceso a memoria.
- ☐ b. Permite ahorrar tres accesos a memoria.
- ☐ c. Permite ahorrar cuatro accesos a memoria.
- ☐ d. Ninguna de las respuestas es correcta.
- ☐ e. Permite ahorrar dos accesos a memoria.

Respuesta incorrecta.

La respuesta correcta es: Permite ahorrar dos accesos a memoria.

Pregunta **12**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/wygC5eEcTrAR56t>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la tubería declarada en la variable fd.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la tubería, imprimirá por consola dos líneas:

```
Leido desde tuberia: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 537

◀ Avisos

Ir a...

[Área personal](#) / [Mis cursos](#) / [Técnicas Digitales III \(Práctica y Exámenes\) 2021](#) / [Parcial 1](#) / [Parcial 1](#)**Comenzado el** martes, 27 de abril de 2021, 21:04**Estado** Finalizado**Finalizado en** martes, 27 de abril de 2021, 21:38**Tiempo  
empleado** 33 minutos 42 segundos**Calificación** 3,302 de 10,000 (33%)Pregunta **1**

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Una la que corresponda, respecto a terminación de un proceso.

Si el proceso intenta acceder por medio de un puntero a una posición fuera de su espacio de direcciones, realiza una

terminación involuntaria por error fatal.



Si el proceso hace una llamada a sistema open() y retorna -1, luego hace una llamada exit(status) con status distinto de 0, realiza una

terminación con error voluntaria.



Si el proceso recibe la señal SIGKILL, realiza una

terminación normal involuntaria.



Si el proceso hace una llamada a sistema return(0) realiza una

terminación por llegada de IPC



Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

La respuesta correcta es: Si el proceso intenta acceder por medio de un puntero a una posición fuera de su espacio de direcciones, realiza una → terminación involuntaria por error fatal., Si el proceso hace una llamada a sistema open() y retorna -1, luego hace una llamada exit(status) con status distinto de 0, realiza una → terminación con error voluntaria., Si el proceso recibe la señal SIGKILL, realiza una → terminación involuntaria por otro proceso., Si el proceso hace una llamada a sistema return(0) realiza una → terminación normal voluntaria.

Pregunta **2**

Incorrecta

Puntúa 0,000 sobre 0,625

Indique lo que crea correcto sobre la syscall kill()

Seleccione una o más de una:

- ☒ a. Siempre se utiliza para terminar un proceso del cual se conoce el PID. ✗
- ☒ b. Se utiliza para configurar ignorar todas las señales a excepción de SIGKILL y SIGSTOP. ✗
- ☐ c. Se utiliza para enviar una señal a un proceso en ejecución, a excepción de SIGKILL y SIGSTOP
- ☐ d. Se puede utilizar para que un proceso se envíe una señal a si mismo.
- ☐ e. Se utiliza para especificar qué hacer cuando el proceso se recibe una señal específica.
- ☐ f. Se utiliza para enviar una señal específica a un proceso en ejecución.
- ☐ g. Se utiliza para especificar qué hacer cuando se recibe la señal SIGKILL.

Respuesta incorrecta.

Las respuestas correctas son: Se puede utilizar para que un proceso se envíe una señal a si mismo., Se utiliza para enviar una señal específica a un proceso en ejecución.

## Pregunta 3

Parcialmente correcta

Puntúa 0,750 sobre 1,250

En el programa hay 3 procesos.

1. El proceso padre crea un hijo (hijo1) el cual queda en una espera activa.
2. Luego el proceso padre crea otro hijo (hijo2) el cual queda en una espera activa.
3. El proceso padre debe enviar la señal SIGKILL a sus dos hijos y ser el único proceso en ejecutar la línea **printf ("Mi pid es %d\n",getpid());**

Completar:

```
int a,b=0;
int main (){
    a= fork();
    //línea en blanco ✓
    if (a==0) {
        while(1);
        signal(SIGKILL,a); ✗
        exit(0);
    }
    b=fork(); ✓
    if (b==0) {
        while(1);
        signal(SIGKILL,b); ✗
        exit(0);
    }
    kill(a,SIGKILL); ✓
    kill(b,SIGKILL);
    printf ("Mi pid es %d\n", getpid());
    sleep(2);
    exit(0);
}
```

wait(NULL);

c=fork();

kill(b,SIGKILL);

Respuesta parcialmente correcta.

Ha seleccionado correctamente 3.

La respuesta correcta es:

En el programa hay 3 procesos.

1. El proceso padre crea un hijo (hijo1) el cual queda en una espera activa.
2. Luego el proceso padre crea otro hijo (hijo2) el cual queda en una espera activa.
3. El proceso padre debe enviar la señal SIGKILL a sus dos hijos y ser el único proceso en ejecutar la línea **printf ("Mi pid es %d\n",getpid());**

Completar:

```
int a,b=0;
int main (){
    a= fork();
    [//linea en blanco]
    if (a==0) {
        while(1);
        [//linea en blanco]
        exit(0);
    }
    [b=fork();]
    if (b==0) {
        while(1);
        [//linea en blanco]
        exit(0);
    }
    [kill(a,SIGKILL);]
    kill(b,SIGKILL);
    printf ("Mi pid es %d\n", getpid());
    sleep(2);
    exit(0);
}
```

Pregunta **4**

Parcialmente correcta

Puntúa 0,208 sobre 0,625

Para la técnica IPC pipe, una lo que corresponda:

Si un proceso intenta escribir en el extremo de escritura de una tubería que cerró todos los descriptores de lectura

da error de lectura y termina.



Si un proceso intenta escribir en el extremo de escritura de una tubería que está vacía

el proceso logra escribir.



Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía y todos sus descriptores de escritura fueron cerrados

el proceso se bloquea.



Si un proceso intenta leer en el extremo de lectura de una tubería que está llena

el proceso lee datos.



Si un proceso intenta escribir en el extremo de escritura de una tubería que está llena

da error de escritura y termina.



Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía

el proceso lee 0 bytes.



Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

La respuesta correcta es: Si un proceso intenta escribir en el extremo de escritura de una tubería que cerró todos los descriptores de lectura → el proceso termina., Si un proceso intenta escribir en el extremo de escritura de una tubería que está vacía → el proceso logra escribir., Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía y todos sus descriptores de escritura fueron cerrados → el proceso lee 0 bytes., Si un proceso intenta leer en el extremo de lectura de una tubería que está llena → el proceso lee datos., Si un proceso intenta escribir en el extremo de escritura de una tubería que está llena → el proceso se bloquea., Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía → el proceso se bloquea.



## Pregunta 5

Parcialmente correcta

Puntúa 0,500 sobre 1,250

Dos procesos relacionados se comunican por medio de una tubería.

1. El proceso hijo lee de la tubería, muestra lo leído y termina.
2. El proceso padre escribe "0123456789" en la tubería, espera a que el proceso hijo termine y luego termina él.

Completar:

```
#define E "0123456789"
int b[2], a, f;
char e[10];
int main (){
    pipe(b);
    a = fork();
    //línea en blanco
    if (a==0){
        write(b[0], e, sizeof(E));
        write(STDOUT_FILENO, e, f);
        exit(0);
    }else {
        f = read(b[0], e, sizeof(e));
        strncpy(e, E, sizeof(E));
        write(b[1], e, sizeof(E));
        close(b[1]);
        exit(0);
    }
}
```

exit(0);

kill(SIGUSR1,b);

pipe(a);

wait(NULL);

read(STDIN\_FILENO, d, sizeof(d));

write(b[0], d, sizeof(d));

Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una tubería.

1. El proceso hijo lee de la tubería, muestra lo leído y termina.
2. El proceso padre escribe "0123456789" en la tubería, espera a que el proceso hijo termine y luego termina él.

Completar:

```
#define E "0123456789"
int b[2], a, f;
char e[10];
int main (){
    [pipe(b);]
    a = fork();
    [//línea en blanco]
    if (a==0){
        [f = read(b[0], e, sizeof(e));]
        write(STDOUT_FILENO, e, f);
        exit(0);
    }else {
        [//línea en blanco]
        strncpy(e, E, sizeof(E));
        write(b[1], e, sizeof(E));
        [wait(NULL);]
        exit(0);
    }
}
```

Pregunta **6**

Sin contestar

Puntúa como 0,625

ordene los pasos en los que se lleva a cabo la llamada a sistema POSIX read()

1	Elegir...
2	Elegir...
3	Elegir...
4	Elegir...
5	Elegir...
6	Elegir...

Respuesta incorrecta.

La respuesta correcta es: 1 → hacer push de los parámetros de la syscall a la pila en orden inverso, 2 → llamada al procedimiento de la biblioteca, 3 → ejecución de la call gate, 4 → ejecución del controlador de periférico de espacio kernel, 5 → retornar al procedimiento de biblioteca de espacio usuario, 6 → retornar del procedimiento de biblioteca al programa de usuario

## Pregunta 7

Parcialmente correcta

Puntúa 0,469 sobre 0,625

Indique lo correcto de la arquitectura IA32

- ☐ a. En modo real solo puede direccionar 20 bits continuos.
- ☒ b. la caché permite aumentar la velocidad de cada acceso a la memoria principal, ya sea de datos o de instrucciones. ✗
- ☒ c. En modo protegido se puede direccionar distintos segmentos de 2<sup>32</sup> bits de largo como máximo. ✓
- ☐ d. Contiene un pipeline de tres etapas que permite como máximo triplicar la ejecución de los programas.
- ☐ e. Siempre es posible ejecutar dos instrucciones a la vez, por ser una arquitectura superescalar.

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

La respuesta correcta es: En modo protegido se puede direccionar distintos segmentos de 2<sup>32</sup> bits de largo como máximo.

## Pregunta 8

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Ordene cronológicamente un cambio de contexto (sin Task Gate).

La instrucción LTR (load task register) carga en la porción visible del TR con el operando.

1	✓
---	---

Se lee la siguiente instrucción a partir del CS e EIP.

6	✓
---	---

Se carga la parte invisible del TR con información del descriptor de TSS.

3	✗
---	---

Se accede al TSS Descriptor especificado en la GDT.

2	✗
---	---

Se guarda el valor de los registros internos en el TSS.

4	✗
---	---

Se carga el valor del TSS a los registros internos.

5	✓
---	---

Respuesta parcialmente correcta.

Ha seleccionado correctamente 3.

La respuesta correcta es: La instrucción LTR (load task register) carga en la porción visible del TR con el operando. → 1, Se lee la siguiente instrucción a partir del CS e EIP. → 6, Se carga la parte invisible del TR con información del descriptor de TSS. → 4, Se accede al TSS Descriptor especificado en la GDT. → 3, Se guarda el valor de los registros internos en el TSS. → 2, Se carga el valor del TSS a los registros internos. → 5

## Pregunta 9

Correcta

Puntúa 0,625 sobre 0,625

Para Protección por Segmentación, indique en que caso(s) tendrá éxito un acceso a datos de otro segmento de acuerdo a los valores de PL.

Seleccione una o más de una:

- ☐ a. DPL=2, CPL=0, RPL=3.
- ☒ b. RPL=1, DPL=2, CPL=0. ✓
- ☐ c. DPL=2, CPL=2, RPL=3.
- ☐ d. RPL=0, CPL=3, DPL=2.
- ☒ e. CPL=0, DPL=2, RPL=2. ✓

## Respuesta correcta

Las respuestas correctas son: RPL=1, DPL=2, CPL=0., CPL=0, DPL=2, RPL=2.

## Pregunta 10

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmeLFJNcP>

2) Modifique el archivo padre.c para que envíe una señal SIGFPE al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGFPE
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 942

## Pregunta 11

Parcialmente correcta

Puntúa 0,125 sobre 0,625

La arquitectura IA32 utiliza numerosos direccionamientos relativos (base / desplazamiento) para procesar una dirección lógica y finalmente obtener una dirección física. Indique el desplazamiento que corresponde a cada base.

Descriptor de segmento	Elegir...	
PTE	Dirección lineal (campo Desplazamiento).	✓
GDTR	Desplazamiento (dirección lógica).	✗
PDE.	Dirección lineal (campo Directorio).	✗
CR3 (PDBR).	Offset (dirección lógica).	✗

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es: Descriptor de segmento → Desplazamiento (dirección lógica), PTE → Dirección lineal (campo Desplazamiento), GDTR → Selector, PDE. → Dirección lineal (campo Tabla), CR3 (PDBR). → Dirección lineal (campo Directorio).

## Pregunta 12

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/W5qgYjCgPLFxRnF>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la FIFO declarada en la variable myfifo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la FIFO, imprimirá por consola dos líneas:

```
Hijo recibe desde FIFO: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:  ✗

La respuesta correcta es: 302

[← Avisos](#)

Ir a...

<b>Comenzado el</b>	martes, 27 de abril de 2021, 21:00
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	martes, 27 de abril de 2021, 21:39
<b>Tiempo empleado</b>	39 minutos 6 segundos
<b>Calificación</b>	<b>3,677</b> de 10,000 ( <b>37%</b> )

Pregunta **1**

Parcialmente correcta

Puntúa 0,063 sobre 0,625

Indique todas las afirmaciones correctas para mecanismos de protección de IA32.

Seleccione una o más de una:

- ☒ a. El chequeo de límite produce una excepción si la dirección a la que se intenta ir es mayor que la base y menor que la base más el límite. ✗
- ☐ b. Esta opción fue eliminada después de que el intento fuera iniciado.
- ☐ c. El chequeo de límite produce una excepción si la dirección a la que se intenta ir es menor que la base o mayor que la base más el límite.
- ☐ d. El chequeo de límite utiliza el bit ED del descriptor para controlar la validez del direccionamiento.
- ☐ e. El chequeo de límite evita a un programa acceder a una región con mayor nivel de privilegio.
- ☒ f. El chequeo de límite evita a un programa acceder a un segmento de distinto tipo. ✗
- ☒ g. El chequeo de límite utiliza el bit G del descriptor para controlar la validez del direccionamiento. ✓

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

Las respuestas correctas son: El chequeo de límite utiliza el bit G del descriptor para controlar la validez del direccionamiento., El chequeo de límite utiliza el bit ED del descriptor para controlar la validez del direccionamiento.

Pregunta **2**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsl>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmeLEJNcP>

2) Modifique el archivo padre.c para que envíe una señal SIGUSR2 al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGUSR2
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 148

Pregunta **3**

Parcialmente correcta

Puntúa 0,104 sobre 0,625

ordene los pasos en los que se lleva a cabo la llamada a sistema POSIX read()

- |   |   |   |
|---|---|---|
| 1 | llamada al procedimiento de la biblioteca                             | ✗ |
| 2 | ejecución de la call gate   | ✗ |
| 3 | ejecución de la task gate   | ✗ |
| 4 | ejecución del controlador de periférico de espacio kernel             | ✓ |
| 5 | retornar del programa de usuario al procedimiento de biblioteca       | ✗ |
| 6 | hacer push de los parámetros de la syscall a la pila en orden inverso | ✗ |

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es: 1 → hacer push de los parámetros de la syscall a la pila en orden inverso, 2 → llamada al procedimiento de la biblioteca, 3 → ejecución de la call gate, 4 → ejecución del controlador de periférico de espacio kernel, 5 → retornar al procedimiento de biblioteca de espacio usuario, 6 → retornar del procedimiento de biblioteca al programa de usuario





Pregunta **4**

Correcta

Puntúa 0,625 sobre 0,625

Una lo que corresponda:

La persistencia de las tuberías	es de proceso.	✓
La persistencia de una cola de Message Queue	es de kernel.	✓
La persistencia de los datos de una FIFO	es de proceso.	✓
La persistencia del nombre de una FIFO	es de sistema de archivos.	✓

Respuesta correcta

La respuesta correcta es: La persistencia de las tuberías → es de proceso., La persistencia de una cola de Message Queue → es de kernel., La persistencia de los datos de una FIFO → es de proceso., La persistencia del nombre de una FIFO → es de sistema de archivos.

Pregunta **5**

Parcialmente correcta

Puntúa 0,500 sobre 0,625

Indique lo que crea correcto sobre la syscall kill()

Seleccione una o más de una:

- ☒ a. Se utiliza para enviar una señal específica a un proceso en ejecución. ✓
- ☐ b. Se utiliza para enviar una señal a un proceso en ejecución, a excepción de SIGKILL y SIGSTOP
- ☐ c. Se utiliza para especificar qué hacer cuando el proceso se recibe una señal específica.
- ☐ d. Se utiliza para manejar todas las señales a excepción de SIGKILL y SIGSTOP.
- ☐ e. Se utiliza para especificar qué hacer cuando se recibe una señal.
- ☒ f. Siempre se utiliza para terminar un proceso poniendo como argumento su PID. ✗
- ☒ g. Se puede utilizar para que un proceso se envíe una señal a si mismo. ✓

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

Las respuestas correctas son: Se puede utilizar para que un proceso se envíe una señal a si mismo., Se utiliza para enviar una señal específica a un proceso en ejecución.



Pregunta **6**

Parcialmente correcta

Puntúa 0,250 sobre 1,250

Dos procesos relacionados se comunican por medio de una tubería.

1-El proceso hijo lee de la tubería, muestra lo leído y termina.

2-El proceso padre escribe "0123456789" en la tubería.

3-El proceso padre espera a que el proceso hijo termine y luego termina él.

Completar:

```
#define E "0123456789"

int b[2], a[1], i;
char j[10];

int main () {
    //linea en blanco
    a[0] = fork();
    wait(NULL);
    if (a[0]==0) {
        close(b[0]);
        i = read(b[0], j, sizeof(j));
        exit(0);
    }
    strncpy(j, E, sizeof(E));
    write(b[1], j, sizeof(E));
    write(STDOUT_FILENO, E, sizeof(E));
    exit(0);
}
```

pipe(b);

i = read(STDIN\_FILENO, j, sizeof(j));

close(b[1]);

i = read(b[0], E, sizeof(E));

pipe(a);

write(STDOUT\_FILENO, j, i);

Respuesta parcialmente correcta.

Ha seleccionado correctamente 0.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una tubería.

1-El proceso hijo lee de la tubería, muestra lo leído y termina.

2-El proceso padre escribe "0123456789" en la tubería.

3-El proceso padre espera a que el proceso hijo termine y luego termina él.

Completar:

```
#define E "0123456789"

int b[2], a[1], i;
char j[10];

int main () {
    [pipe(b);]
    a[0] = fork();
    [//linea en blanco]
    if (a[0]==0) {
        [i = read(b[0], j, sizeof(j));]
        [write(STDOUT_FILENO, j, i);]
        exit(0);
    }
    strncpy(j, E, sizeof(E));
    write(b[1], j, sizeof(E));
```



```
[wait(NULL);]  
exit(0);  
}
```



Pregunta **7**

Correcta

Puntúa 1,250 sobre 1,250

En el siguiente programa se ejecutan 3 procesos.

1. El proceso padre crea un proceso hijo (hijo1). El proceso hijo queda en una espera activa.
2. Luego el proceso padre crea otro proceso hijo (hijo2). El proceso hijo queda en una espera activa.
3. El proceso padre envía la señal SIGKILL a los procesos hijos y espera a que los mismos terminen.
4. La línea **printf ("Mi pid es %d\n", getpid());** la debe ejecutar solo el padre.

Completar:

```
int c,e=0;
int main (){
     ✓
    if (c==0) {
         ✓
        while(1);
        exit(0);
    }
    e = fork();
     ✓
    if (e==0) {
        while(1);
        exit(0);
    }
    kill(c, SIGKILL);
     ✓
    wait(NULL);
    wait(NULL);
    printf ("Mi pid es %d\n", getpid());
    exit(0);
}
```

Respuesta correcta

La respuesta correcta es:

En el siguiente programa se ejecutan 3 procesos.

1. El proceso padre crea un proceso hijo (hijo1). El proceso hijo queda en una espera activa.
2. Luego el proceso padre crea otro proceso hijo (hijo2). El proceso hijo queda en una espera activa.
3. El proceso padre envía la señal SIGKILL a los procesos hijos y espera a que los mismos terminen.
4. La línea **printf ("Mi pid es %d\n", getpid());** la debe ejecutar solo el padre.

Completar:



```
int c,e=0;
int main (){
    [c = fork();]
    if (c==0) {
        [//línea en blanco]
        while(1);
        exit(0);
    }
    e = fork();
    [//línea en blanco]
    if (e==0) {
        while(1);
        exit(0);
    }
    kill(c, SIGKILL);
    [kill(e, SIGKILL);]
    wait(NULL);
    wait(NULL);
    printf ("Mi pid es %d\n", getpid());
    exit(0);
}
```

Pregunta **8**

Incorrecta

Puntúa 0,000 sobre 0,625

Indique cuál de las siguientes afirmaciones es correcta referidas a la abstracción Procesos:

Seleccione una o más de una:

- ☐ a. Cuando los procesos son IObounded, se logra una gran mejora en el uso de CPU, usando multiprogramación .
- ☒ b. No se justifica hacer multiprogramación cuando los procesos son interactivos. ✗
- ☒ c. Cada programa tiene un espacio de direcciones independiente del resto. ✗
- ☐ d. La abstracción de procesos permite ejecutar en pseudo-paralelo procesos.
- ☐ e. La multiprogramación es la conmutación entre programas de manera rápida, salvando su estado en las conmutaciones.
- ☒ f. Pueden coexistir dos programas con un mismo proceso. ✗

Respuesta incorrecta.

Las respuestas correctas son: La abstracción de procesos permite ejecutar en pseudo-paralelo procesos., Cuando los procesos son IObounded, se logra una gran mejora en el uso de CPU, usando multiprogramación .



Pregunta **9**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsl>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/W5qgYjCgPLExRnF>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la FIFO declarada en la variable myfifo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la FIFO, imprimirá por consola dos líneas:

```
Hijo recibe desde FIFO: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 302

Pregunta **10**

Parcialmente correcta

Puntúa 0,260 sobre 0,625

Indique todas las afirmaciones verdaderas en la traslación de direcciones lineales a físicas para la IA-32:

Seleccione una o más de una:

- ☐ a. Cada Page Table está compuesta por 1024 entradas de 1 bytes cada una.
- ☒ b. Los bits 0-11 de la dirección lineal se usan offset para del marco de página. ✓
- ☒ c. Los bits de privilegio de la PDE y PTE indican en que nivel de privilegio se está trabajando. ✗
- ☐ d. Los bits 12-21 de la dirección lineal se usan como base de la Page Table.
- ☒ e. El tamaño del marco de página no depende del bit granularidad del descriptor del segmento usado. ✓
- ☐ f. El Page Directory tiene 1024 entradas de 4 bytes y siempre ocupa la misma región de memoria.
- ☐ g. Los bits 22-31 de la dirección lineal se usan para encontrar el PDE en la Page Directory.

Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

Las respuestas correctas son: Los bits 22-31 de la dirección lineal se usan para encontrar el PDE en la Page Directory., El tamaño del marco de página no depende del bit granularidad del descriptor del segmento usado., Los bits 0-11 de la dirección lineal se usan offset para del marco de página.



Pregunta **11**

Correcta

Puntúa 0,625 sobre 0,625

El campo Back Link del TSS de la tarea actual, contiene el selector de segmento de TSS de la tarea anterior. Este campo permite que se inicie un cambio de tarea a la tarea anterior cuando se ejecuta una instrucción JMP, CALL o IRET.

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

Pregunta **12**

Incorrecta

Puntúa 0,000 sobre 0,625

Para una arquitectura IA32, se intenta acceder a memoria física y existe un acierto de TLB. Indique cuál afirmación es correcta (comparada con un fallo de TLB):

Seleccione una:

- ☒ a. Permite ahorrar cuatro accesos a memoria.
- ☐ b. Permite ahorrar un acceso a memoria.
- ☐ c. Permite ahorrar dos accesos a memoria.
- ☐ d. Permite ahorrar tres accesos a memoria.
- ☐ e. Ninguna de las respuestas es correcta.

✗

Respuesta incorrecta.

La respuesta correcta es: Permite ahorrar dos accesos a memoria.

◀ Avisos

Ir a...



[Área personal](#) / [Mis cursos](#) / [Técnicas Digitales III \(Práctica y Exámenes\) 2021](#) / [Parcial 1](#) / [Parcial 1](#)

**Comenzado el** martes, 27 de abril de 2021, 21:00

**Estado** Finalizado

**Finalizado en** martes, 27 de abril de 2021, 21:31

**Tiempo  
empleado** 31 minutos 26 segundos

**Calificación** 6,000 de 10,000 (60%)

Pregunta **1**

Correcta

Puntúa 0,625  
sobre 0,625

Seleccione lo correcto respecto a señales

Seleccione una:

- ☐ a. Las señales se generan exclusivamente con la syscall kill().
- ☐ b. Se pueden enviar señales entre procesos no relacionados, a excepción de SIGSTOP.
- ☒ c. Una señal se usa generalmente para comunicar estados entre procesos.
- ☐ d. Un proceso puede bloquear la recepción de todas las señales.
- ☐ e. Una señal es la petición por parte del usuario para la terminación de un proceso.



Respuesta correcta

La respuesta correcta es: Una señal se usa generalmente para comunicar estados entre procesos.



Pregunta **2**

Sin contestar

Puntúa como  
1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/wygC5eEcTrAR56t>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la tubería declarada en la variable fd.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la tubería, imprimirá por consola dos líneas:

```
Leido desde tuberia: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 537

Pregunta **3**

Incorrecta

Puntúa 0,000  
sobre 0,625

Para Protección por Segmentación, indique en que caso(s) tendrá éxito un acceso a datos de otro segmento de acuerdo a los valores de PL.

Seleccione una o más de una:

- ☒ a. CPL=0, DPL=2, RPL=2.
- ☒ b. DPL=2, CPL=2, RPL=3.
- ☐ c. RPL=0, CPL=3, DPL=2.
- ☐ d. RPL=1, DPL=2, CPL=0.
- ☒ e. DPL=2, CPL=0, RPL=3.



Respuesta incorrecta.

Las respuestas correctas son: RPL=1, DPL=2, CPL=0., CPL=0, DPL=2, RPL=2.

Pregunta **4**

Correcta

Puntúa 1,250  
sobre 1,250

En el siguiente programa se ejecutan 3 procesos.

1. El proceso padre crea un proceso hijo (hijo1). El proceso hijo queda en una espera activa.
2. Luego el proceso padre crea otro proceso hijo (hijo2). El proceso hijo queda en una espera activa.
3. El proceso padre envía la señal SIGKILL a los procesos hijos y espera a que los mismos terminen.
4. La línea **printf ("Mi pid es %d\n", getpid());** la debe ejecutar solo el padre.

Completar:

```
int c,e=0;
int main (){
    c = fork();
    if (c==0) {
        //línea en blanco
        while(1);
        exit(0);
    }
    e = fork();
    if (e==0) {
        while(1);
        exit(0);
    }
    kill(c, SIGKILL);
    wait(NULL);
    wait(NULL);
    printf ("Mi pid es %d\n", getpid());
    exit(0);
}
```

Respuesta correcta

La respuesta correcta es:

En el siguiente programa se ejecutan 3 procesos.

1. El proceso padre crea un proceso hijo (hijo1). El proceso hijo queda en una espera activa.
2. Luego el proceso padre crea otro proceso hijo (hijo2). El proceso hijo queda en una espera activa.
3. El proceso padre envía la señal SIGKILL a los procesos hijos y espera a que los mismos terminen.
4. La línea **printf ("Mi pid es %d\n", getpid());** la debe ejecutar solo el padre.

Completar:

```
int c,e=0;
int main (){
    [c = fork();]
    if (c==0) {
        [//línea en blanco]
        while(1);
        exit(0);
    }
    e = fork();
    [//línea en blanco]
    if (e==0) {
        while(1);
        exit(0);
    }
    kill(c, SIGKILL);
    [kill(e, SIGKILL);]
    wait(NULL);
    wait(NULL);
    printf ("Mi pid es %d\n", getpid());
    exit(0);
}
```

Pregunta **5**

Correcta

Puntúa 1,250  
sobre 1,250

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo abre la FIFO con permisos de solo escritura, escribe "0123456789" en la FIFO, cierra y elimina la FIFO y luego termina él.
2. El proceso padre abre la FIFO con permisos de solo lectura, lee la FIFO y muestra lo leído.
3. El proceso padre espera a que el proceso hijo termine para cerrar la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define AA "/tmp/BB"
int a,b,c,e;
char d[10];
int main(){
    mkfifo(AA, 0777);
    b=fork();
    if(b==0){
        //linea en blanco ✓
        c = open(AA, O_WRONLY, 0);
        write(c, E,sizeof(E)); ✓
        close(c);
        unlink(AA);
        exit(0);
    }
    a = open(AA, O_RDONLY, 0); ✓
    e = read(a, d, sizeof(d));
    write(STDOUT_FILENO, d, e);
    wait(NULL); ✓
    close(a); ✓
    exit(0);
}
```

```
close(c);
```

```
kill(SIGUSR1,x);
```

```
write(a, d, sizeof(d));
```

```
read(c, E,sizeof(E));
```

Respuesta correcta

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo abre la FIFO con permisos de solo escritura, escribe "0123456789" en la FIFO, cierra y elimina la FIFO y luego termina él.
2. El proceso padre abre la FIFO con permisos de solo lectura, lee la FIFO y muestra lo leído.
3. El proceso padre espera a que el proceso hijo termine para cerrar la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define AA "/tmp/BB"
int a,b,c,e;
char d[10];
int main(){
    mkfifo(AA, 0777);
    b=fork();
    if(b==0){
        [//línea en blanco]
        c = open(AA, O_WRONLY, 0);
        [write(c, E,sizeof(E));]
        close(c);
        unlink(AA);
        exit(0);
    }
    [a = open(AA, O_RDONLY, 0);]
    e = read(a, d, sizeof(d));
    write(STDOUT_FILENO, d, e);
    [wait(NULL);]
    [close(a);]
    exit(0);
}
```

Pregunta **6**  
Parcialmente  
correcta  
Puntúa 0,427  
sobre 0,625

Para la técnica IPC pipe, una lo que corresponda:

Si un proceso intenta escribir en el extremo de escritura de una tubería que está llena

el proceso se bloquea.



Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía y todos sus descriptores de escritura fueron cerrados

da error de lectura y termina.



Si un proceso intenta leer en el extremo de lectura de una tubería que está llena

el proceso lee datos.



Si un proceso intenta escribir en el extremo de escritura de una tubería que cerró todos los descriptores de lectura

da error de escritura y termina.



Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía

el proceso se bloquea.



Si un proceso intenta escribir en el extremo de escritura de una tubería que está vacía

el proceso logra escribir.



Respuesta parcialmente correcta.

Ha seleccionado correctamente 4.

La respuesta correcta es: Si un proceso intenta escribir en el extremo de escritura de una tubería que está llena → el proceso se bloquea., Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía y todos sus descriptores de escritura fueron cerrados → el proceso lee 0 bytes., Si un proceso intenta leer en el extremo de lectura de una tubería que está llena → el proceso lee datos., Si un proceso intenta escribir en el extremo de escritura de una tubería que cerró todos los descriptores de lectura → el proceso termina., Si un proceso intenta leer en el extremo de lectura de una tubería que está vacía → el proceso se bloquea., Si un proceso intenta escribir en el extremo de escritura de una tubería que está vacía → el proceso logra escribir.

Comentario:



Pregunta **7**

Incorrecta

Puntúa 0,000  
sobre 0,625

Para una arquitectura IA32, indique cuál de las siguientes definiciones es **falsa**:

Seleccione una:



- ☐ a. La TLB es una caché de traslación de direcciones lógicas a físicas.
- ☒ b. Un acierto de TLB permite ahorrar dos accesos a direcciones lógicas. ✗
- ☐ c. En modo protegido, es obligatorio al menos el uso de segmentación para trasladar direcciones.
- ☐ d. Un acierto de TLB permite ahorrar dos accesos a memoria física.
- ☐ e. La parte oculta de los registros de segmento permite ahorrar un acceso a memoria (mientras no se cambie de segmento).

Respuesta incorrecta.

La respuesta correcta es: La TLB es una caché de traslación de direcciones lógicas a físicas.

Pregunta **8**  
Parcialmente  
correcta  
Puntúa 0,417  
sobre 0,625

Seleccione lo correcto para tipos de estructura de sistemas operativos

- ☒ a. Los OS cliente/servidor son una generalización de OS microkernel . 
- ☐ b. Los OS Máquina Virtual se basan en organizar las funciones del OS de manera jerárquica.
- ☒ c. Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers. 
- ☐ d. Los OS cliente/servidor se ejecutan mas rápido que los OS monolíticos.
- ☐ e. Los OS cliente/servidor se caracterizan por separar multiprogramación de máquina extendida.
- ☐ f. Hasta los OS monolíticos pueden tener una estructura interna.

Respuesta parcialmente correcta.






Ha seleccionado correctamente 2.

Las respuestas correctas son: Hasta los OS monolíticos pueden tener una estructura interna., Los OS cliente/servidor son una generalización de OS microkernel ., Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers.

Pregunta **9**  
Parcialmente  
correcta  
Puntúa 0,156  
sobre 0,625

Indique todas las afirmaciones verdaderas en la traslación de direcciones lógicas a lineales para Intel IA-32:

Seleccione una o más de una:

- ☒ a. Para poder acceder a una LDT es necesario encontrar el LDT descriptor en la GDT. 
- ☐ b. Si el bit TI indica que es una LDT, se utiliza el GDTR para acceder a la LDT.
- ☒ c. El GDTR es de 48 bits. 
- ☒ d. Los bits 3-15 del selector siempre se utilizan para acceder a la GDT. 
- ☒ e. Todos los descriptores de LDT están almacenados fuera de la GDT. 
- ☒ f. El bit de granularidad de un descriptor permite seleccionar bloques de 4 Kbytes o de 1 Byte. 

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

Las respuestas correctas son: Para poder acceder a una LDT es necesario encontrar el LDT descriptor en la GDT., El GDTR es de 48 bits.

Pregunta **10**

Correcta

Puntúa 1,250  
sobre 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/W5qgYjCgPLFxRnF>

2) Modifique el archivo padre.c para que envíe una señal SIGFPE al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGFPE
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 981

Pregunta **11**

Incorrecta

Puntúa 0,000  
sobre 0,625

El campo Back Link del TSS de la tarea actual, contiene el selector de segmento de TSS de la tarea anterior. Este campo permite que se inicie un cambio de tarea a la tarea anterior cuando se ejecuta una instrucción IRET.

Seleccione una:

- ☐ Verdadero
- ☒ Falso **✗**

La respuesta correcta es 'Verdadero'

Pregunta **12**

Correcta

Puntúa 0,625  
sobre 0,625

Indique que información se guarda en cada entrada de la tabla de procesos (PCB):

Seleccione una o más de una:

- ☐ a. PID del hijo.
- ☐ b. ID de los hilos que está usando.
- ☒ c. Descriptores de archivos abiertos. **✓**
- ☐ d. IPC que usa actualmente el proceso.
- ☒ e. El valor del contador de programa. **✓**

Respuesta correcta

Las respuestas correctas son: El valor del contador de programa., Descriptores de archivos abiertos.

[◀ Avisos](#)[Ir a...](#)



<b>Comenzado el</b>	martes, 27 de abril de 2021, 21:00
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	martes, 27 de abril de 2021, 21:33
<b>Tiempo empleado</b>	33 minutos 14 segundos
<b>Calificación</b>	<b>3,458</b> de 10,000 (35%)

Pregunta **1**

Incorrecta

Puntúa 0,000 sobre 0,625

Para una arquitectura IA32, se quiere acceder a memoria física, y da fallo de TLB. indique cuál afirmación es correcta:

Seleccione una:

- ☐ a. Se deben hacer dos accesos adicionales a memoria, comparado a si hubiera acierto de TLB.
- ☐ b. Se debe hacer un acceso adicional a memoria, comparado a si hubiera acierto de TLB.
- ☐ c. Se deben hacer cuatro accesos adicionales a memoria, comparado a si hubiera acierto de TLB.
- ☒ d. Ninguna de las respuestas es correcta.
- ☐ e. Se deben hacer tres accesos adicionales a memoria, comparado a si hubiera acierto de TLB.



Respuesta incorrecta.

La respuesta correcta es: Se deben hacer dos accesos adicionales a memoria, comparado a si hubiera acierto de TLB.

Pregunta **2**

Parcialmente correcta

Puntúa 0,625 sobre 1,250

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d,e=0;
7
8 int main (){
9     a = fork();
10    b = fork();
11    printf ("1-Mi pid es %d\n", getpid());
12    if (a==0) {
13        c = fork();
14        exit(0);
15    }
16    printf ("2-Mi pid es %d\n", getpid());
17    if (b==0) {
18        exit(0);
19    }
20    printf ("3-Mi pid es %d\n", getpid());
21    d = fork();
22    if (d==0) {
23        exit(0);
24    }
25    printf ("4-Mi pid es %d\n", getpid());
26    sleep(2);
27    exit(0);
28 }
29

```

Colocar la cantidad de procesos que ejecutan las líneas

11:  ✓

16:  ✗

20:  ✗

25:  ✓

Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

La respuesta correcta es:





```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d,e=0;
7
8 int main (){
9     a = fork();
10    b = fork();
11    printf ("1-Mi pid es %d\n", getpid());
12    if (a==0) {
13        c = fork();
14        exit(0);
15    }
16    printf ("2-Mi pid es %d\n", getpid());
17    if (b==0) {
18        exit(0);
19    }
20    printf ("3-Mi pid es %d\n", getpid());
21    d = fork();
22    if (d==0) {
23        exit(0);
24    }
25    printf ("4-Mi pid es %d\n", getpid());
26    sleep(2);
27    exit(0);
28 }
29

```

Colocar la cantidad de procesos que ejecutan las líneas

11: [4]

16: [2]

20: [1]

25: [1]

### Pregunta 3

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnSL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmelFJNCp>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la FIFO declarada en la variable myfifo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la FIFO, imprimirá por consola dos líneas:

```
Hijo recibe desde FIFO: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 114



Pregunta **4**

Correcta

Puntúa 0,625 sobre 0,625

Para Protección por Segmentación, indique en que caso(s) no podrá hacer un acceso a datos de otro segmento de acuerdo a los valores de PL.

Seleccione una o más de una:

- ☐ a. DPL=3, CPL=1, RPL=2.
- ☒ b. RPL=3, DPL=2, CPL=0.
- ☒ c. CPL=3, DPL=2, RPL=1.
- ☐ d. DPL=2, RPL=1, CPL=1.
- ☐ e. RPL=2, CPL=1, DPL=2.



Respuesta correcta

Las respuestas correctas son: RPL=3, DPL=2, CPL=0., CPL=3, DPL=2, RPL=1.

Pregunta **5**

Incorrecta

Puntúa 0,000 sobre 0,625

El campo Back Link del TSS de la tarea actual, contiene el selector de segmento de TSS de la tarea anterior. Este campo permite que se inicie un cambio de tarea a la tarea anterior cuando se ejecuta una instrucción JMP, CALL o IRET.

Seleccione una:

- ☒ Verdadero
- ☐ Falso

La respuesta correcta es 'Falso'




Pregunta **6**

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Indique lo que crea correcto sobre la syscall kill()

Seleccione una o más de una:

- ☐ a. Siempre se utiliza para terminar un proceso del cual se conoce el PID.
- ☐ b. Se utiliza para enviar una señal a un proceso en ejecución, a excepción de SIGKILL y SIGSTOP
- ☐ c. Se puede utilizar para que un proceso se envíe una señal a si mismo.
- ☒ d. Se utiliza para enviar una señal específica a un proceso en ejecución. 
- ☐ e. Se utiliza para especificar qué hacer cuando se recibe la señal SIGKILL.
- ☐ f. Se utiliza para especificar qué hacer cuando el proceso se recibe una señal específica.
- ☐ g. Se utiliza para configurar ignorar todas las señales a excepción de SIGKILL y SIGSTOP.

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: Se puede utilizar para que un proceso se envíe una señal a si mismo., Se utiliza para enviar una señal específica a un proceso en ejecución.



Pregunta **7**

Parcialmente correcta

Puntúa 0,750 sobre 1,250

Dos procesos relacionados se comunican por medio de una tubería.

1. El proceso hijo escribe "0123456789" en la tubería y termina.
2. El proceso padre lee de la tubería y muestra lo leído.

Completar:

```
#define E "0123456789"
int a[2], b,e;
char d[10];
int main (){
    pipe(a); ✓
    b = fork(); ✓
    if (b==0){
        close(a[0]); ✗
        close(a[0]);
        strncpy(d, E, sizeof(E) );
        write(a[0], d, sizeof(d)); ✗
        exit(0);
    }
    close(a[1]);
    e = read(a[0], d, sizeof(d));
    write(STDOUT_FILENO, d, e); ✓
    exit(0);
}
```

pipe(b);

write(STDOUT\_FILENO, E, sizeof(E));

close(a[1]);

wait(NULL);

write(a[1], d, sizeof(E));

//línea en blanco

read(a[1], d, sizeof(d));

Respuesta parcialmente correcta.

Ha seleccionado correctamente 3.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una tubería.

1. El proceso hijo escribe "0123456789" en la tubería y termina.
2. El proceso padre lee de la tubería y muestra lo leído.

Completar:



```
#define E "0123456789"
int a[2], b,e;
char d[10];
int main (){
    [pipe(a);]
    [b = fork();]
    if (b==0){
        [//línea en blanco]
        close(a[0]);
        strncpy(d, E, sizeof(E) );
        [write(a[1], d, sizeof(E));]
        exit(0);
    }
    close(a[1]);
    e = read(a[0], d, sizeof(d));
    [write(STDOUT_FILENO, d, e);]
    exit(0);
}
```

Pregunta **8**

Incorrecta

Puntúa 0,000 sobre 0,625

Indique todas las afirmaciones verdaderas en la traslación de direcciones lógicas a lineales para Intel IA-32:

Seleccione una o más de una:

- ☒ a. Los bits 3-15 del selector siempre se utilizan para acceder a la GDT. ✗
- ☒ b. Para poder acceder a una LDT es necesario encontrar el LDT descriptor en la GDT. ✓
- ☐ c. Si el bit TI indica que es una LDT, se utiliza el GDTR para acceder a la LDT.
- ☐ d. El bit de granularidad de un descriptor permite seleccionar bloques de 4 Kbytes o de 1 Byte.
- ☐ e. El GDTR es de 48 bits.
- ☒ f. Todos los descriptors de LDT están almacenados fuera de la GDT. ✗

Respuesta incorrecta.

Las respuestas correctas son: Para poder acceder a una LDT es necesario encontrar el LDT descriptor en la GDT., El GDTR es de 48 bits.



## Pregunta 9

Parcialmente correcta

Puntúa 0,521 sobre 0,625

Una lo que corresponda.

El cambio de estado de Ready a Running es debido a	el planificador del sistema operativo.	✓
El cambio de estado de Blocked a Running es debido a	no es válida esa transición.	✓
El cambio de estado de Blocked a Ready es debido a	interrupción de hardware producida por un periférico.	✓
El cambio de estado de Ready a Finished es debido a	no es válida esa transición.	✓
El cambio de estado de Running a Ready es debido a	interrupción de hardware producida por un timer.	✓
El cambio de estado de Running a Blocked es debido a	interrupción de hardware producida por un periférico.	✗

Respuesta parcialmente correcta.

Ha seleccionado correctamente 5.

La respuesta correcta es: El cambio de estado de Ready a Running es debido a → el planificador del sistema operativo., El cambio de estado de Blocked a Running es debido a → no es válida esa transición., El cambio de estado de Blocked a Ready es debido a → interrupción de hardware producida por un periférico., El cambio de estado de Ready a Finished es debido a → no es válida esa transición., El cambio de estado de Running a Ready es debido a → interrupción de hardware producida por un timer., El cambio de estado de Running a Blocked es debido a → la ejecución de syscall bloqueante.

## Pregunta 10

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/W5qgYjCgPLFxRnF>

2) Modifique el archivo padre.c para que envíe una señal SIGUSR1 al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGUSR1
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:  ✗

La respuesta correcta es: 154



Pregunta **11**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione lo correcto para tipos de estructura de sistemas operativos

- ☐ a. Los OS cliente/servidor se caracterizan por separar multiprogramación de máquina extendida.
- ☐ b. Hasta los OS monolíticos pueden tener una estructura interna.
- ☐ c. Los OS cliente/servidor son una generalización de OS microkernel .
- ☒ d. Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers. ✓
- ☒ e. Los OS cliente/servidor se ejecutan mas rápido que los OS monolíticos. ✗
- ☒ f. Los OS Máquina Virtual se basan en organizar las funciones del OS de manera jerárquica. ✗

Respuesta incorrecta.

Las respuestas correctas son: Hasta los OS monolíticos pueden tener una estructura interna., Los OS cliente/servidor son una generalización de OS microkernel ., Los OS microkernel son mas robustos que los OS monolíticos en cuanto a fallas de drivers.

Pregunta **12**

Correcta

Puntúa 0,625 sobre 0,625

La syscall mkfifo():

Seleccione una:

- ☐ a. Permite especificar que sea no bloqueante.
- ☐ b. Permite borrar una fifo.
- ☒ c. Permite crear una fifo. ✓
- ☐ d. Retorna el tamaño de la fifo, si no hubo error.
- ☐ e. Se debe usar en conjunto con la llamada fork().
- ☐ f. Retorna el descriptor, si no hubo error.
- ☐ g. Permite abrir una fifo en modo bloqueante o no bloqueante.
- ☐ h. Tiene como único argumento el nombre del fifo.

Respuesta correcta

La respuesta correcta es: Permite crear una fifo.

◀ Avisos

Ir a...



<b>Comenzado el</b>	martes, 27 de abril de 2021, 21:00
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	martes, 27 de abril de 2021, 21:40
<b>Tiempo empleado</b>	39 minutos 51 segundos
<b>Calificación</b>	<b>3,646</b> de 10,000 ( <b>36%</b> )

Pregunta **1**

Parcialmente correcta

Puntúa 0,208 sobre 0,625

Indique cuál de las siguientes alternativas es válida para hacer un cambio de tarea:

Seleccione una o más de una:

- ☐ a. La tarea actual ejecuta una instrucción JMP o CALL a un descriptor TSS en la GDT.
- ☐ b. La tarea actual ejecuta una instrucción JMP o CALL con el bit NT en 0 del registro EFLAGS
- ☐ c. La tarea actual ejecuta una instrucción IRET con el bit NT en 1 del registro EFLAGS.
- ☐ d. La tarea ejecuta una instrucción JMP o CALL a un descriptor TSS en la LDT.
- ☐ e. La tarea actual ejecuta una instrucción JMP o CALL a un descriptor de Call Gate en la GDT actual.
- ☒ f. La tarea actual ejecuta una instrucción JMP o CALL a un descriptor de Task Gate en la GDT actual.
- ☐ g. Una interrupción apunta a un descriptor de TSS en la IDT.



Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: La tarea actual ejecuta una instrucción JMP o CALL a un descriptor TSS en la GDT., La tarea actual ejecuta una instrucción JMP o CALL a un descriptor de Task Gate en la GDT actual., La tarea actual ejecuta una instrucción IRET con el bit NT en 1 del registro EFLAGS.



Pregunta **2**

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Indique cuál de las siguientes afirmaciones es correcta cuando un proceso hijo (B) termina antes que su padre (A):

Seleccione una o más de una:

- ☐ a. El proceso A se convierte en un proceso huérfano.
- ☒ b. Se libera la memoria principal que se usaba en el proceso B.
- ☐ c. El proceso A finaliza al recibir la señal SIGCHLD.
- ☐ d. El proceso A es adoptado por el proceso init
- ☐ e. El proceso A se convierte en un proceso zombie.
- ☐ f. El proceso B se convierte en un proceso huérfano.
- ☐ g. El proceso B se convierte en un proceso zombie.



Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: El proceso B se convierte en un proceso zombie., Se libera la memoria principal que se usaba en el proceso B.

Pregunta **3**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsl>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmeLFJNcP>

2) Modifique el archivo padre.c para que envíe una señal SIGUSR2 al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGUSR2
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 148

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo lee de la FIFO, muestra lo leído, cierra la FIFO y luego termina él.
2. El proceso padre escribe "0123456789" en la FIFO, espera a que el proceso hijo termine.
3. El proceso padre para cierra y elimina la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define CC "/tmp/BB"
int e,b,f,g;
char d[10];
int main(){
    mkfifo(CC, 0777);
    b=fork();
    if(b>0){
        //linea en blanco
        f = open(CC, O_WRONLY, 0);
        write(f, E,sizeof(E));
        wait(NULL);
        close(f);
        unlink(CC);
        exit(0);
    }
    e = open(CC, O_RDONLY, 0);
    write(STDOUT_FILENO, d, g);
    exit(0);
}
```

close(e);

kill(SIGUSR1,x);

close(f)

exit(0);

f = open(CC, O\_RDWR, 0);

read(e, d, sizeof(d));

g = read(e, d, sizeof(d));

read(f, E,sizeof(E));

pipe(e);

g = write(e, d, sizeof(d));

Respuesta parcialmente correcta.

Ha seleccionado correctamente 5.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso hijo lee de la FIFO, muestra lo leído, cierra la FIFO y luego termina él.
2. El proceso padre escribe "0123456789" en la FIFO, espera a que el proceso hijo termine.
3. El proceso padre para cierra y elimina la FIFO y luego terminar él.

Completar:

```
#define E "0123456789"
#define CC "/tmp/BB"
int e,b,f,g;
char d[10];
int main(){
    mkfifo(CC, 0777);
    b=fork();
    if(b>0){
        [//linea en blanco]
        f = open(CC, O_WRONLY, 0);
        [write(f, E,sizeof(E));]
        wait(NULL);
        close(f);
        [unlink(CC);]
        exit(0);
    }
    [//linea en blanco]
    e = open(CC, O_RDONLY, 0);
    [g = read(e, d, sizeof(d));]
    write(STDOUT_FILENO, d, g);
    [close(e);]
    exit(0);
}
```

Pregunta 5

Parcialmente correcta

Puntúa 0,313 sobre 1,250

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d=0;
7
8 int main (){
9     a = fork();
10    b = fork();
11    c = fork();
12    if (a==0) { |
13        exit(0);
14        printf ("1-Mi pid es %d\n", getpid());
15    }
16    printf ("2-Mi pid es %d\n", getpid());
17    if (b==0) {
18        exit(0);
19    }
20    d=fork();
21    printf ("3-Mi pid es %d\n", getpid());
22    if (c>0) {
23        sleep(2);
24        exit(0);
25    }
26    printf ("4-Mi pid es %d\n", getpid());
27    exit(0);
28 }
29
```

Colocar la cantidad de procesos que ejecutan las líneas

14:  ❌

16:  ✅

21:  ❌

26:  ❌

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d=0;
7
8 int main (){
9     a = fork();
10    b = fork();
11    c = fork();
12    if (a==0) { |
13        exit(0);
14        printf ("1-Mi pid es %d\n", getpid());
15    }
16    printf ("2-Mi pid es %d\n", getpid());
17    if (b==0) {
18        exit(0);
19    }
20    d=fork();
21    printf ("3-Mi pid es %d\n", getpid());
22    if (c>0) {
23        sleep(2);
24        exit(0);
25    }
26    printf ("4-Mi pid es %d\n", getpid());
27    exit(0);
28 }
29

```

Colocar la cantidad de procesos que ejecutan las líneas

14: [0]

16: [4]

21: [4]

26: [2]

#### Pregunta 6

Parcialmente correcta

Puntúa 0,313 sobre 0,625

Indique cuál de las siguientes transiciones de estados de un proceso NO es válida.

Seleccione una o más de una:

- ☐ a. Running a Finished.
- ☐ b. Ready a Running.
- ☒ c. Ready a Finished.
- ☐ d. Block a Running.
- ☐ e. Blocked a Ready.
- ☐ f. Running a Blocked.
- ☐ g. Running a Ready.
- ☐ h. New a Ready.



Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: Ready a Finished., Block a Running.

Pregunta **7**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/5LT7b5GmeLFJNcP>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la FIFO declarada en la variable myfifo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la FIFO, imprimirá por consola dos líneas:

```
Hijo recibe desde FIFO: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 114

Pregunta **8**

Correcta

Puntúa 0,625 sobre 0,625

Para Protección por Segmentación, indique en que caso(s) tendrá éxito un acceso a datos de otro segmento de acuerdo a los valores de PL.

Seleccione una o más de una:

☐ a. RPL=0, CPL=3, DPL=2.

☒ b. RPL=1, DPL=2, CPL=0.



☐ c. DPL=2, CPL=0, RPL=3.

☒ d. CPL=0, DPL=2, RPL=2.



☐ e. DPL=2, CPL=2, RPL=3.

Respuesta correcta

Las respuestas correctas son: RPL=1, DPL=2, CPL=0., CPL=0, DPL=2, RPL=2.

Pregunta **9**

Parcialmente correcta

Puntúa 0,469 sobre 0,625

Indique la o las afirmaciones correctas respecto de la IPC FIFO:

Seleccione una o más de una:

- ☐ a. Se usan para enviar y recibir mensajes.
- ☐ b. Se destruyen y eliminan al cerrar los descriptores.
- ☒ c. Poseen dos descriptores, uno de lectura y otro de escritura.
- ☐ d. Se utilizan solo para enviar datos entre procesos no relacionados.
- ☒ e. Se pueden abrir en forma bloqueante y no bloqueantes.
- ☒ f. Es obligatorio el uso de un nombre en el sistema de archivos.

✗

✓

✓

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

Las respuestas correctas son: Se pueden abrir en forma bloqueante y no bloqueantes., Es obligatorio el uso de un nombre en el sistema de archivos.

Pregunta **10**

Parcialmente correcta

Puntúa 0,260 sobre 0,625

Indique todas las afirmaciones verdaderas en la traslación de direcciones lineales a físicas para la IA-32:

Seleccione una o más de una:

- ☐ a. Cada Page Table está compuesta por 1024 entradas de 1 bytes cada una.
- ☒ b. Los bits 22-31 de la dirección lineal se usan para encontrar el PDE en la Page Directory.
- ☒ c. Los bits 12-21 de la dirección lineal se usan como base de la Page Table.
- ☐ d. El Page Directory tiene 1024 entradas de 4 bytes y siempre ocupa la misma región de memoria.
- ☐ e. El tamaño del marco de página no depende del bit granularidad del descriptor del segmento usado.
- ☐ f. Los bits de privilegio de la PDE y PTE indican en que nivel de privilegio se está trabajando.
- ☒ g. Los bits 0-11 de la dirección lineal se usan offset para del marco de página.

✓

✗

✓

Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

Las respuestas correctas son: Los bits 22-31 de la dirección lineal se usan para encontrar el PDE en la Page Directory., El tamaño del marco de página no depende del bit granularidad del descriptor del segmento usado., Los bits 0-11 de la dirección lineal se usan offset para del marco de página.

Pregunta **11**

Incorrecta

Puntúa 0,000 sobre 0,625

Seleccione lo correcto respecto a señales

Seleccione una:

- ☒ a. Las señales se generan por una interrupción debida a un evento externo.
- ☐ b. Una señal es una notificación entregada a un proceso debido a un evento asíncrono.
- ☐ c. Una señal es la petición por parte del usuario para la creación de un nuevo proceso.
- ☐ d. Un proceso puede bloquear la recepción de todas las señales.
- ☐ e. Solo se pueden enviar señales entre procesos relacionados (padre-hijo-nieto).



Respuesta incorrecta.

La respuesta correcta es: Una señal es una notificación entregada a un proceso debido a un evento asíncrono.

Pregunta **12**

Parcialmente correcta

Puntúa 0,104 sobre 0,625

Indique lo correcto de la arquitectura IA32

- ☒ a. No siempre es posible ejecutar dos instrucciones a la vez, por mas que sea una arquitectura superescalar.
- ☐ b. En modo protegido se puede direccionar solo un segmento de  $2^{32}$  bits de largo como máximo.
- ☒ c. la caché permite aumentar la velocidad de cada acceso a la memoria principal, ya sea de datos o de instrucciones.
- ☐ d. Contiene dos pipelines de tres etapas que permite como máximo triplicar la ejecución de los programas.
- ☐ e. En modo real solo puede direccionar 20 bits segmentados en bloques de 64Kb .



Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: No siempre es posible ejecutar dos instrucciones a la vez, por mas que sea una arquitectura superescalar., En modo real solo puede direccionar 20 bits segmentados en bloques de 64Kb .

◀ Avisos

Ir a...



<b>Comenzado el</b>	martes, 27 de abril de 2021, 21:01
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	martes, 27 de abril de 2021, 21:40
<b>Tiempo empleado</b>	38 minutos 43 segundos
<b>Calificación</b>	<b>1,844</b> de 10,000 ( <b>18%</b> )

## Pregunta 1

Parcialmente correcta

Puntúa 0,313 sobre 1,250

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d=0;
7
8 int main (){
9     a= fork();
10    b= fork();
11    c= fork();
12    printf ("1-Mi pid es %d\n", getpid());
13    if (b==0) {
14        exit(0);
15    }
16    printf ("2-Mi pid es %d\n", getpid());
17    d= fork();
18    if (a==0) {
19        exit(0);
20    }
21    printf ("3-Mi pid es %d\n", getpid());
22    if (c==0) {
23        exit(0);
24    }
25    printf ("4-Mi pid es %d\n", getpid());
26    exit(0);
27 }
28
```

Colocar la cantidad de procesos que ejecutan las líneas

12:  ✖16:  ✔21:  ✖25:  ✖   

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <sys/types.h>
4 #include <unistd.h>
5
6 int a,b,c,d=0;
7
8 int main (){
9     a= fork();
10    b= fork();
11    c= fork();
12    printf ("1-Mi pid es %d\n", getpid());
13    if (b==0) {
14        exit(0);
15    }
16    printf ("2-Mi pid es %d\n", getpid());
17    d= fork();
18    if (a==0) {
19        exit(0);
20    }
21    printf ("3-Mi pid es %d\n", getpid());
22    if (c==0) {
23        exit(0);
24    }
25    printf ("4-Mi pid es %d\n", getpid());
26    exit(0);
27 }
28

```

Colocar la cantidad de procesos que ejecutan las líneas

12: [8]

16: [4]

21: [4]

25: [2]

## Pregunta 2

Parcialmente correcta

Puntúa 0,188 sobre 0,625

Indique todas las afirmaciones correctas para mecanismos de protección de IA32.

Seleccione una o más de una:

- ☒ a. En la paginación, el nivel 0 de privilegios es el de mayor privilegio. ✗
- ☐ b. La paginación cuenta con protección por chequeo de Límite.
- ☐ c. La segmentación cuenta con protección por tipo de página.
- ☒ d. La protección por privilegios previene a un programa acceder a un segmento de mayor privilegio. ✓
- ☐ e. La paginación cuenta con chequeo de límite.
- ☐ f. Las task gates permiten acceder a un nivel de privilegio mayor.
- ☐ g. La paginación cuenta con protección por dominio direccionable.

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: La protección por privilegios previene a un programa acceder a un segmento de mayor privilegio., La paginación cuenta con protección por dominio direccionable.

Pregunta **3**

Parcialmente correcta

Puntúa 0,375 sobre 0,625

Indique lo que crea correcto sobre la syscall kill()

Seleccione una o más de una:

- ☒ a. Se utiliza para enviar una señal específica a un proceso en ejecución. ✓
- ☒ b. Se puede utilizar para que un proceso se envíe una señal a si mismo. ✓
- ☒ c. Se utiliza para enviar una señal a un proceso en ejecución, a excepción de SIGKILL y SIGSTOP ✗
- ☐ d. Se utiliza para especificar qué hacer cuando el proceso se recibe una señal específica.
- ☒ e. Siempre se utiliza para terminar un proceso del cual se conoce el PID. ✗
- ☐ f. Se utiliza para configurar ignorar todas las señales a excepción de SIGKILL y SIGSTOP.
- ☐ g. Se utiliza para especificar qué hacer cuando se recibe la señal SIGKILL.

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

Las respuestas correctas son: Se puede utilizar para que un proceso se envíe una señal a si mismo., Se utiliza para enviar una señal específica a un proceso en ejecución.

Pregunta **4**

Parcialmente correcta

Puntúa 0,156 sobre 0,625

Una la que corresponda, respecto a terminación de un proceso.

- Si el proceso intenta acceder por medio de un puntero a una posición fuera de su espacio de direcciones, realiza una 

terminación involuntaria por otro proceso. ✗
- Si el proceso hace una llamada a sistema return(0) realiza una 

terminación normal voluntaria. ✓
- Si el proceso hace una llamada a sistema open() y retorna -1, luego hace una llamada exit(status) con status distinto de 0, realiza una 

terminación involuntaria por error fatal. ✗
- Si el proceso recibe la señal SIGKILL, realiza una 

terminación por llegada de IPC ✗

Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

La respuesta correcta es: Si el proceso intenta acceder por medio de un puntero a una posición fuera de su espacio de direcciones, realiza una → terminación involuntaria por error fatal., Si el proceso hace una llamada a sistema return(0) realiza una → terminación normal voluntaria., Si el proceso hace una llamada a sistema open() y retorna -1, luego hace una llamada exit(status) con status distinto de 0, realiza una → terminación con error voluntaria., Si el proceso recibe la señal SIGKILL, realiza una → terminación involuntaria por otro proceso.

Pregunta **5**

Parcialmente correcta

Puntúa 0,469 sobre 0,625

Indique lo correcto de la arquitectura IA32

- ☒ a. En modo protegido se puede direccionar distintos segmentos de  $2^{32}$  bits de largo como máximo. ✓
- ☐ b. En modo real solo puede direccionar 20 bits continuos.
- ☐ c. Contiene un pipeline de tres etapas que permite como máximo triplicar la ejecución de los programas.
- ☐ d. Siempre es posible ejecutar dos instrucciones a la vez, por ser una arquitectura superescalar.
- ☒ e. la caché permite aumentar la velocidad de cada acceso a la memoria principal, ya sea de datos o de instrucciones. ✗

Respuesta parcialmente correcta.

Ha seleccionado demasiadas opciones.

La respuesta correcta es: En modo protegido se puede direccionar distintos segmentos de  $2^{32}$  bits de largo como máximo.

Pregunta **6**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnSL>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/W5qgYjCgPLExRnE>

2) Modifique el archivo padre.c para que envíe una señal SIGFPE al proceso hijo.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) Cuando el proceso hijo reciba la señal indicada, imprimirá por consola dos líneas:

```
Hijo recibe SIGFPE
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 981

Indique todas las afirmaciones verdaderas en la traslación de direcciones lineales a físicas para la IA-32:

Seleccione una o más de una:

- ☒ a. Los bits 12-21 de la dirección lineal se usan como base de la Page Table.
- ☐ b. Los bits de privilegio de la PDE y PTE indican en que nivel de privilegio se está trabajando.
- ☐ c. Cada Page Table está compuesta por 1024 entradas de 1 bytes cada una.
- ☒ d. Los bits 0-11 de la dirección lineal se usan offset para del marco de página.
- ☐ e. El Page Directory tiene 1024 entradas de 4 bytes y siempre ocupa la misma región de memoria.
- ☒ f. Los bits 22-31 de la dirección lineal se usan para encontrar el PDE en la Page Directory.
- ☐ g. El tamaño del marco de página no depende del bit granularidad del descriptor del segmento usado.

Respuesta parcialmente correcta.

Ha seleccionado correctamente 2.

Las respuestas correctas son: Los bits 22-31 de la dirección lineal se usan para encontrar el PDE en la Page Directory., El tamaño del marco de página no depende del bit granularidad del descriptor del segmento usado., Los bits 0-11 de la dirección lineal se usan offset para del marco de página.

Ordene cronológicamente un cambio de contexto (sin Task Gate).

- Se carga el valor del TSS a los registros internos.

2

✖
- Se lee la siguiente instrucción a partir del CS e EIP.

4

✖
- Se guarda el valor de los registros internos en el TSS.

3

✖
- Se accede al TSS Descriptor especificado en la GDT.

1

✖
- La instrucción LTR (load task register) carga en la porción visible del TR con el operando.

5

✖
- Se carga la parte invisible del TR con información del descriptor de TSS.

6

✖

Respuesta incorrecta.

La respuesta correcta es: Se carga el valor del TSS a los registros internos. → 5, Se lee la siguiente instrucción a partir del CS e EIP. → 6, Se guarda el valor de los registros internos en el TSS. → 2, Se accede al TSS Descriptor especificado en la GDT. → 3, La instrucción LTR (load task register) carga en la porción visible del TR con el operando. → 1, Se carga la parte invisible del TR con información del descriptor de TSS. → 4

Pregunta **9**

Sin contestar

Puntúa como 1,250

1) Descargue los siguientes 2 archivos en un mismo directorio:

padre.c : <https://nube.ingenieria.uncuyo.edu.ar/s/gbxbSWbePfrRnsl>

hijo.o : <https://nube.ingenieria.uncuyo.edu.ar/s/wygC5eEcTrAR56t>

2) Modifique el archivo padre.c para que envíe al proceso hijo el contenido de la variable tx\_buffer mediante la tubería declarada en la variable fd.

3) Ambos archivos se compilan con el siguiente comando en consola:

```
> gcc -c padre.c && gcc -O0 hijo.o padre.o -o padre
```

4) El binario generado por el comando anterior se ejecuta en consola con:

```
> ./padre
```

5) El proceso hijo cuando reciba el contenido de tx\_buffer por la tubería, imprimirá por consola dos líneas:

```
Leído desde tubería: HOLA HIJO MIO
```

```
El resultado del ejercicio es: XXX
```

6) Copie el número de 3 cifras XXX en la casilla de abajo. Este número es la respuesta del ejercicio.

Respuesta:



La respuesta correcta es: 537

Pregunta **10**

Parcialmente correcta

Puntúa 0,083 sobre 0,625

Seleccione las afirmaciones correctas respecto a Posix Pipes:

Seleccione una o más de una:

- ☒ a. Permiten el intercambio de datos entre procesos relacionados.
- ☐ b. Se destruyen al cerrar los descriptores de lectura.
- ☐ c. Poseen persistencia de kernel.
- ☐ d. Se pueden abrir en forma bloqueantes y no bloqueantes.
- ☒ e. Poseen un nombre en el sistema de archivos.
- ☐ f. Permiten sincronizar procesos relacionados.
- ☐ g. Se destruyen al cerrar todos los descriptores de lectura y escritura.
- ☐ h. Se utilizan para enviar datos entre procesos no relacionados.



Respuesta parcialmente correcta.

Ha seleccionado correctamente 1.

Las respuestas correctas son: Se destruyen al cerrar todos los descriptores de lectura y escritura., Permiten sincronizar procesos relacionados., Permiten el intercambio de datos entre procesos relacionados.

Pregunta **11**

Incorrecta

Puntúa 0,000 sobre 0,625

Indique cuál de las siguientes afirmaciones es correcta cuando un proceso hijo (B) termina antes que su padre (A):

Seleccione una o más de una:

- ☐ a. El proceso A se convierte en un proceso zombie.
- ☐ b. El proceso A se convierte en un proceso huérfano.
- ☐ c. El proceso B se convierte en un proceso zombie.
- ☐ d. El proceso A es adoptado por el proceso init
- ☐ e. Se libera la memoria principal que se usaba en el proceso B.
- ☐ f. El proceso A finaliza al recibir la señal SIGCHLD.
- ☒ g. El proceso B se convierte en un proceso huérfano.



Respuesta incorrecta.

Las respuestas correctas son: El proceso B se convierte en un proceso zombie., Se libera la memoria principal que se usaba en el proceso B.



Pregunta **12**

Incorrecta

Puntúa 0,000 sobre 1,250

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso padre abre la FIFO con permisos de solo lectura, lee la FIFO y muestra lo leído.
2. El proceso padre espera a que el proceso hijo termine para cerrar y eliminar la FIFO y luego termina él.
3. El proceso hijo abre la FIFO con permisos de solo escritura, escribe "0123456789" en la FIFO, cierra la FIFO y luego termina él.

Completar:

```
#define E "0123456789"
#define AA "/tmp/CC"
int a,x,c,f;
char d[10];
int main(){
    mkfifo(AA, 0777);
     ✖
    x=fork();
    switch (x) {
        case 0:
             ✖
            write(c, E,sizeof(E));
            close(c);
            exit(0);
            break;
        default:
             ✖
            f = read(a, d, sizeof(d));
            write(STDOUT_FILENO, d, f);
            break;
    }
     ✖
    close(a);
     ✖
    exit(0);
}
```

Respuesta incorrecta.

La respuesta correcta es:

Dos procesos relacionados se comunican por medio de una FIFO.

1. El proceso padre abre la FIFO con permisos de solo lectura, lee la FIFO y muestra lo leído.
2. El proceso padre espera a que el proceso hijo termine para cerrar y eliminar la FIFO y luego termina él.
3. El proceso hijo abre la FIFO con permisos de solo escritura, escribe "0123456789" en la FIFO, cierra la FIFO y luego termina él.

Completar:

```
#define E "0123456789"
#define AA "/tmp/CC"
int a,x,c,f;
char d[10];
int main(){
    mkfifo(AA, 0777);
    [//linea en blanco]
    x=fork();
    switch (x) {
        case 0:
            [c = open(AA, O_WRONLY , 0);]
            write(c, E,sizeof(E));
            close(c);
            exit(0);
        break;
        default:
            [a = open(AA, O_RDONLY, 0);]
            f = read(a, d, sizeof(d));
            write(STDOUT_FILENO, d, f);
        break;
    }
    [wait(NULL);]
    close(a);
    [unlink(AA);]
    exit(0);
}
```

[◀ Avisos](#)

Ir a...