

Ejercicios Parcial 1

LCC

2021

1. Introducción

Ejercicio 1

Indicar con cual/cuales de los conceptos de SO (abstracción, aislamiento o administración) se vinculan: 1) sockets, 2) memoria virtual, 3) malloc, 4) senales, 5) gestion de permisos y usuarios, 6) bloqueos, 7) archivos.

Ejercicio 2

Indicar 2 razones por las que estudiar sistemas operativos desde el punto de vista de un desarrollador y 2 desde el punto de vista de un administrador.

Ejercicio 3

Dar algun ejemplo de problema del mundo real que pueda ser resuelto por algún algoritmo visto en SOI.

Ejercicio 4

En qué difiere NachOS de un SO de propósito general? Qué arquitectura de CPU simula? de qué tipo es? Qué registros tiene? Qué otras características se destacan?

Ejercicio 5

Enunciar las diferencias entre SO de propósito general y SO para dispositivos móviles.
Indicar verdadero o falso:

- La mayor parte de los sistemas operativos modernos tiene una arquitectura basada en microkernels
- Un sistema operativo con organización híbrida mantiene algunas partes del SO en el espacio de usuario
- Un sistema operativo con organización monolítica mantiene algunas partes del SO en el espacio de usuario
- Un sistema operativo organizado como un microkernel mantiene la mayor parte del SO dentro del kernel
- El uso de la memoria virtual se hizo necesario cuando aparecieron las computadoras de uso personal
- En los sistemas multiprogramados fue necesario definir prioridades a los procesos

Ejercicio 6

Qué arquitectura (ARM o Intel) y qué SO se utiliza mayoritariamente para 1) servidores, 2) propósito general en empresas, oficina y computadoras personales, 3) dispositivos móviles?

Ejercicio 7

Marcar la respuesta correcta. A diferencia de la multitarea cooperativa, la apropiativa...

- Es inmune a que un cálculo demasiado largo o un ciclo infinito dejen a la computadora efectivamente congelada.
- Es la más utilizada hoy en día.
- Ocurre sólo cuando el proceso hace una llamada al sistema.
- Se emplea principalmente en sistemas multiusuario.
- Requiere apoyo de hardware.

Ejercicio 8

Indicar cuales características (abstracción, aislamiento, administración, ninguna) se corresponde con las siguientes afirmaciones

- Instrumentar políticas que repartan la atención del sistema de forma efectiva y acorde a las necesidades establecidas entre los diferentes procesos o usuarios.
- Cada proceso y cada usuario no tendrán que preocuparse por otros que estén usando el mismo sistema; idealmente, su experiencia será la misma que si el sistema estuviera exclusivamente dedicado a su atención. Requiere que el sistema operativo cuente con ayuda del hardware.
- Presentar una interfaz consistente al usuario (puede ser gráfica o textual), eliminando las diferencias que provendrían de manejar distintos tipos de hardware.
- Los programadores no deben tener que preocuparse de los detalles del acceso a hardware, o de la configuración particular de una computadora. El programador debe poder enfocarse en resolver los problemas o necesidades particulares de sus usuarios.

Ejercicio 9

Un programa típicamente pasa por varias etapas en su ejecución, algunas de las cuales están limitadas por el procesador, mientras que las otras lo están por la entrada/salida. Los componentes del sistema que están ocupados en cada caso son distintos. ¿Qué tipo de sistemas nacieron para responder a esta necesidad?

Ejercicio 10

Se presentó que los sistemas microkernel se basan en la simplificación de los mecanismos de comunicación y un esquema más claro de comunicación entre componentes. Sin embargo, los sistemas monolíticos siempre fueron más simples de implementar, razón por la cual muchos sistemas microkernel se han reducido a ejercicios académicos. Explique esta tensión.

Ejercicio 11

De la siguiente lista de eventos, indicar cuáles corresponden a interrupciones y cuáles a excepciones

- El usuario accionó alguna tecla
- Se produjo un acceso ilegal a memoria fuera del segmento (segmentation fault)
- El proceso en ejecución lanzó una llamada al sistema
- Llegó un paquete a la interfaz de red
- Se produjo una división sobre cero
- El proceso en ejecución estuvo activo ya demasiado tiempo, es hora de un cambio de contexto
- Los datos solicitados al controlador de disco duro ya están disponibles

Ejercicio 12

Indicar con qué se corresponden los siguientes conceptos (abstracción, aislamiento o administración)

- espacio libre en disco
- uso de cpu
- permisos de archivos
- memoria virtual

Ejercicio 13

Una de las principales funciones de un sistema operativo es la administración de recursos. Explicar y mencionar un ejemplo.

Ejercicio 14

Explicar la diferencia entre tarea apropiativa y cooperativa

2. Administración de procesos

Ejercicio 1

Indicar verdadero o falso

- Un programa puede asociarse con varios procesos.
- Varios programas diferentes pueden dar lugar a un proceso.
- Tarea y programa son conceptos similares.

Ejercicio 2

En un momento dado sólo se pueden estar ejecutando un número de instrucciones menor o igual a la cantidad de procesadores del sistema. Cómo es posible que en un sistema con un solo procesador se pueda navegar por inet, compilar un programa y escuchar música al mismo tiempo?

Ejercicio 3

Indicar V/F

- Un proceso en estado listo puede pasar directamente al estado zombie/terminado
- Un proceso en ejecución puede pasar a bloqueado sólo en llamadas al sistema
- Un proceso siempre se bloquea en algún momento

Ejercicio 4

Indicar verdadero o falso (sobre nachOS)

- Thread::Sleep pone el hilo en estado listo.
- Thread::Yield pone el hilo en estado listo.
- Si no se utiliza la opción “-rs” y un proceso hace Yield entonces no será el siguiente en ejecutarse.
- Si se utiliza la opción “-rs” y un proceso hace Yield entonces no será el siguiente en ejecutarse.
- Si un proceso llama a Signal::Wait entonces pasará por el estado bloqueado.

3. Planificación de procesos

Ejercicio 1

En un sistema interactivo, los procesos típicamente están en ejecución un largo periodo (entre minutos y días), sin embargo, cuando se comparan algoritmos de planificación estos son casi siempre tratados como procesos cortos. ¿Qué significa esto? ¿Cuál sería un ejemplo de proceso largo?

Ejercicio 2

Proceso	Llegada	t
A	0	8
B	2	13
C	4	3
D	4	6
E	6	8
F	6	3

Desarrollar la representación gráfica de cómo el dispatcher asignaría la CPU a los procesos en (??) bajo

- FIFO
- Ronda con $q = 1$
- Ronda con $q = 3$
- SPN
- Retroalimentación multinivel con $q = 1$, $n = 1$ e incremento nq

También incluir la tabla de análisis. Cuáles son las ventajas y desventajas de éstos esquemas?

Ejercicio 3

La política de multicolos con prioridad con dos clases de prioridad puede ser bastante parecida a la ronda egoísta. ¿En qué casos se comportarán igual ambas políticas?

Ejercicio 4

Dos formas de mejorar las multicolos con prioridad son degradar a los procesos que recibieron mucha atención (aging) y promover a los procesos que recibieron poca atención (“retroalimentación inversa”). Comparar las dos alternativas, indicando ventajas y desventajas.

3.1. Ejercicio 5

Utilizar el comando `time` para ejecutar algún programa o comando que ejecute durante más de un segundo. Indicar, justificando su respuesta, cuáles pueden ser los valores máximos para:

- Tiempo de núcleo
- Tiempo de sistema
- Tiempo de usuario
- Tiempo de uso de procesador
- Tiempo desocupado
- Utilización del CPU

Cuando sea posible, también indicar los valores mínimos.

Ejercicio 6

CFS intenta hacer una multitarea ideal y precisa. ¿Con qué objetivo(s) de planificación de los mencionados en el libro no es compatible esta idea tal como se la presenta en el apunte? ¿Cómo es que este planificador “beneficia” a los procesos interactivos?

Ejercicio 7

Se deben ejecutar los siguientes procesos (0 es la mayor prioridad)

Proceso	Prioridad	Llegada	t
A	1	0	8
B	2	2	13
C	1	4	3
D	0	4	6

Desarrollar la representación gráfica de cómo el despachador les asignaría el CPU y la tabla de análisis para las siguientes políticas de planificación.

- Multicolos con prioridad (sin retroalimentación) y $q = 2$ (para todas las colas)
- Multicolos con prioridad (sin retroalimentación) y $q = 2, 4, 8, 16$ para las colas de prioridad 0, 1, 2, 3 respectivamente
- Multicolos con prioridad (sin retroalimentación) y $q = 2$ (para todas las colas). Si un proceso consume 2 veces su quantum antes de bloquearse (se asume que los procesos nunca se bloquean) entonces es degradado a una cola inferior (tantas veces como sea necesario, hasta la cola de prioridad 3).

Considerar que:

- Un proceso que expira su quantum se agrega al final de la cola correspondiente pero antes de los procesos que llegan en ese tick
- Si hay empate, desempata orden alfabético
- Si aparece un proceso de prioridad mayor al que se está ejecutando, deberá esperar a que finalice el quantum del proceso en ejecución

Ejercicio 8

Verdadero o falso: Agrandar el quantum en Round Robin beneficia los programas interactivos.

Ejercicio 9

Calcule el tiempo de respuesta para cada proceso si se utiliza multicolos con prioridad con FIFO dentro de cada cola de prioridad. Suponga que los tiempos de núcleo y del planificador son despreciables y que los procesos no se bloquean. Los procesos tienen prioridad fija (A=2, B=10, C=5, D=5) siendo 10 la mayor prioridad. Cuando un proceso de mayor prioridad está listo interrumpe la ejecución de los de baja prioridad instantáneamente.

Proceso	Llegada	t
A	1	2
B	4	4
C	0	8
D	5	4

Repetir el ejercicio para Round robin $q = 2$ dentro de cada cola.

Ejercicio 10

Explicar qué problemas surgen en un sistema de multicolos con prioridad y distintas formas de resolverlo.

Ejercicio 11

Indicar verdadero o falso justificando la respuesta.

- HPRN incluye información intrínseca de los procesos
- Si se utiliza selfish round robin y el ritmo de los procesos nuevos es menor que el ritmo de los procesos aceptados, entonces el algoritmo se comporta como FIFO

Ejercicio 12

Realizar un esquema de la planificación y hacer la tabla de análisis para un sistema utiliza ronda egoísta (selfish round robin) con quantum $q = 1$ para planificar sus procesos. Los siguientes procesos están en la cola de procesos nuevos:

Procesos	t	Prioridad
A	2	1
B	2	2

Los siguientes procesos están en la cola de aceptados:

Procesos	t	Prioridad
C	3	5
D	4	7

Teniendo en cuenta que el ritmo de incremento de las colas es $a = 2$ para la cola de procesos nuevos (la prioridad se incrementa en 2 en cada tick) y $b = 1$ para la cola de procesos aceptados. Considerar que un proceso que es aceptado se integra a la cola de aceptados antes de poner en la cola al proceso que acaba de utilizar su quantum. Al comienzo desempate por orden alfabético (suponga que C está primero en la cola).

Y si A comenzara con prioridad 0?

Ejercicio 13

Hacer la tabla de análisis para los siguientes procesos utilizando planificación en ronda con un quantum de 2 ticks

Procesos	Llegada	t
A	0	4
B	1	5
C	5	4

Ejercicio 14

Hacer la tabla de análisis para los siguientes procesos utilizando planificación SPN con un quantum de 2 ticks

Procesos	Llegada	t
A	0	3
B	0	2
C	1	2

Ejercicio 15

os procesos A, B y C son despachados en el siguiente orden: A(2 ticks), B(1 tick), A(1 tick), C(2 ticks)

Si C terminó, están listos A y B y se utiliza SPN (en un sistema no apropiativo, pero colaborativo) calcular e'_A y e'_B e indicar qué proceso será elegido a continuación (notar que A se ejecutó dos veces). Utilice $f = 0,9$ y e inicial = 1.

Ejercicio 15

Podemos decir que Linux utiliza un algoritmo por cola dentro de FB. Explique. ¿por qué se utilizan distintas políticas dentro de algunas colas?

Ejercicio 16

- Entre Round Robin y FIFO. Cuál conviene usar para un sistema interactivo?
- Entre SPN y Round Robin. Cuál conviene usar para un sistema no interactivo?

Ejercicio 17

Planificación en Linux. Indicar V o F

- Todos los procesos listos tienen estado TASK_RUNNING
- La prioridad estática se establece con nice o set_priority
- El nuevo CFS del kernel 2.6.24 utiliza un árbol ordenado por rq-¿fairlock - vruntime

Ejercicio 18

Se tiene un sistema que utiliza multicolos con prioridad para planificar procesos

- No se usa envejecimiento ni promoción
- La cola 0 es la de mayor prioridad y usa Round Robin con $q = 2$
- La cola 1 usa Round Robin con $q = 4$
- Si varios procesos están listos se usa el orden alfabético para desempatar
- Un proceso de alta prioridad no puede quitarle el procesador a uno de baja prioridad hasta que finalice su quantum

Proceso	Llegada	t	Prioridad
A	0	5	0
B	0	3	0
C	0	3	1
D	0	3	1
E	3	1	1
F	9	4	0

Realice la representación gráfica y la tabla de análisis.

Ejercicio 19

Explicar el problema de inversión de prioridades y diferentes mecanismos para solucionarlo

Ejercicio 20

Comentar modificaciones a la política básica de multicolos

Ejercicio 21

El planificador CFS del kernel 2.6.24 incluye soporte para multiprocesadores y planificación de grupos. Describir brevemente estas mejoras.

Ejercicio 22

¿Cuáles de los algoritmos estudiados son más susceptibles a la inanición que se presenta cuando $\rho > 1$ (saturación)?
¿Cuáles menos? Identifique por lo menos dos en cada caso.

Ejercicio 23

Sobre el algoritmo de planificación lotería:

- ¿Para qué tipo de carga es más apto y menos apto?
- ¿Qué tan susceptible resulta a producir inanición?
- ¿Qué tan justa sería su ejecución?
- ¿Qué modificaciones requeriría para planificar procesos con necesidades de tiempo real?

Ejercicio 24

Tanto la afinidad a procesador como el balanceo de cargas son elementos importantes y deseables en todo planificador que se ejecute en un entorno multiprocesador. Sin embargo, afinidad y balanceo de cargas trabajan uno en contra del otro. Explicar la anterior afirmación, y elaborar cuándo debe predominar cada uno de estos mecanismos.

4. Administración de memoria y senales

Ejercicio 1

Indicar verdadero o falso

- Con kill se puede hacer lo mismo que permite hacer raise
- Con raise se puede hacer lo mismo que permite hacer kill
- signal envía una senal
- Si un proceso recibe la senal SIGSEGV es porque se produjo un fallo de segmentacion

Ejercicio 2

Un programa con un fallo de segmentación podría capturar SIGSEGV. ¿Qué ocurre en ese caso si el handler no hace más que imprimir un cartel y retornar?

Ejercicio 3

Explicar la diferencia entre el intercambio total y parcial

Ejercicio 4

Aplicar las políticas FIFO, OPT y LRU considerando una traza con 4 páginas físicas

1 2 3 4 1 2 5 1 2 3 4 5

Ejercicio 5

Diagrame el acomodo del espacio en memoria de un proceso. ¿Qué diferencias principales y qué similitudes tienen la sección de datos con el espacio de libres (“heap”), y el espacio de libres con la pila (“stack”)?

Ejercicio 6

Un proceso en un sistema con arquitectura de memoria basada en segmentación tiene la siguiente tabla de segmentos

Segmento	Inicio	Tamano	Permisos
0	240	600	rx
1	2300	16	r
2	90	100	rw
3	1320	950	rw
4	-	96	rx

Para cada una de las siguiente solicitudes, indicar qué dirección física correspondería y -de ser el caso- qué excepción se genera.

- (a) Lectura, 0-430
- (b) Escritura, 0-150
- (c) Lectura, 1-15
- (d) Escritura, 2-130
- (e) Ejecución, 4-25

Ejercicio 7

El buffer de traducción adelantada (TLB, “translation lookaside buffer”) de un sistema en particular presenta una efectividad de 95 %. Obtener un valor del TLB toma 10 ns. La memoria principal tarda 120 ns en recuperar un valor. ¿Cuál es el tiempo promedio para completar una operación a memoria?

Ejercicio 8

Con la siguiente cadena de referencia y empleando 4 marcos de referencia de memoria física, desarrollar la asignación bajo esquemas FIFO, OPT y LRU.

1, 3, 2, 1, 5, 3, 4, 1, 5, 2, 6, 7, 5, 7, 2, 5, 3, 5, 3, 1

Asumiendo que cada fallo de página toma 8 ms en ser atendido, ¿qué diferencia en rendimiento puede observarse?

Ejercicio 9

Suponga un sistema paginado con un rango de direcciones de 4 GiB (4 294 967 296 direcciones)

- Cuántas páginas tendrá el sistema si se utilizan 4096 bits?
- Qué tamaño en bits tendrá una entrada de la tabla de traducción? Suponer que solo se guarda el número de marco físico
- Qué tamaño tendrá la tabla de paginación si se desea cubrir todo el rango?
- Suponga que el tamaño de la tabla fuera demasiado grande. Proponer 2 soluciones explicando ventajas y desventajas de cada una.

Ejercicio 10

Explicar la relación entre direcciones virtuales y físicas. Indicar cómo se realiza la traducción en cada caso:

- Una computadora con TLB y tabla de paginación de un nivel con la entrada cargada en la TLB
- Una computadora con TLB y tabla de paginación de un nivel sin la entrada cargada en la TLB
- Una computadora sin TLB y tabla de paginación de dos niveles
- Una computadora sin paginación ni TLB pero con segmentación

Ejercicio 11

Un equipo presenta rendimiento muy deficiente. Ante un análisis de utilización a lo largo de un día, se encuentra que el promedio de uso de CPU está a 20 % y el uso de la interfaz al disco duro que a la memoria virtual a 95 %. ¿En qué condición está el sistema?

Elegir de la siguiente lista las 2 respuestas que mayor efectividad tendrían para mejorar el rendimiento del sistema. Justificar.

1. Reemplazar la CPU por una 20 % más rápida (pasar de 1 GHz a 1,2 GHz).
2. Destinar un área del disco 25 % más grande para la memoria virtual (pasar de 4 a 5 GiB)
3. Aumentar el grado de multiprogramación en 10 % (aumentar de 20 a 22 los procesos en la cola de ejecución)
4. Reducir el grado de multiprogramación en 10 % (reducir de 20 a 18 los procesos en cola de ejecución)
5. Instalar un 25 % más de memoria principal (pasar de 2 a 2.5 GiB)
6. Instalar un disco duro un 33 % más rápido (cambiar un disco de 5400 RPM por uno de 7200 RPM)

Ejercicio 12

Explicar la diferencia entre SIG_IGN y SIG_DFL. para qué sirve el valor retornado por la función signal?

Ejercicio 13

Explicar 3 ventajas de usar sigaction en vez de las senales tradicionales de UNIX

Ejercicio 14

Dado el siguiente programa

```
#include <sys/types.h>
#include <signal.h>
#include <unistd.h>

main(){
    raise(SIGCHLD);
    printf("Despues_del_raise\n");
    return 0;
}
```

1. Modificarlo para que tenga el mismo efecto pero sin utilizar la función raise
2. Qué ocurre cuando se ejecuta el programa? termina normalmente?
3. Modificarlo para que muestre un mensaje con el número de la senal recibida cuando se reciba la senal. En este caso, cómo se completa la ejecución del programa después de imprimir el mensaje?

Ejercicio 15

Una máquina utiliza direcciones de 16 bits. Se deben alojar en memoria 2 programas: A que requiere 4300 bytes y B que requiere 3068 bytes de memoria. Si se utiliza paginación con páginas de 128 bytes y cada entrada en la tabla de paginación ocupa 8 bytes, responder

1. Cuánto espacio se desperdicia por fragmentación interna y externa en cada programa?
2. Describa el funcionamiento de las tablas de paginación de 2 niveles. Qué ventajas tiene este modelo? Suponiendo que se usan 3 bits para el primer nivel, indicar (utilizando valores ficticios) cómo se traduciría la dirección virtual 0x1234 utilizando este modelo

Ejercicio 16

Explique una de las desventajas de utilizar tablas de paginación de dos (o más niveles). ¿Cómo se puede mitigar el problema?

Ejercicio 17

¿Cuál es la diferencia entre fragmentación externa y fragmentación interna? ¿Qué problemas trae cada una?

Ejercicio 18

Hacer el análisis completo de la siguiente traza usando la política FIFO considerando un sistema con 5 páginas físicas

1 7 2 4 5 3 5 1 3 5 6

Ejercicio 19

Completar el siguiente fragmento de forma de que sleep1 se comporte como sleep(int n) que suspende un proceso por n segundos (suponer que no llegan senales durante sleep)

```
int bool = 1;

static void sig_alrm(int signo) {
    bool = 0;
}

unsigned int sleep1(unsigned int nsecs) {
    signal(..., sig_alrm);
    ...
    while(bool);
    bool = 1;
    return(alarm(0));
}
```

Ejercicio 20

Se ejecuta el siguiente programa

```
#include <signal.h>
#include <stdio.h>

void h(int arg){
    printf("fpe\n");
}

main(){
    signal(SIGFPE, h);
    int a = 1/0;
}
```

Responder verdadero o falso justificando

- El proceso nunca recibirá SIGSEGV
- El código signal(SIGFPE, h) hace que el proceso reciba la senal SIGFPE
- El proceso nunca emitiría una senal a otro proceso
- El proceso terminará su ejecución si recibe cualquier otra senal