Sistemas Operativos – Grado Ingeniera Informática UDC. Enero 2016
Sólo puede usar lápiz, bolígrafo y calculadora.
Tiempo máximo para todo el examen: 3h

Parte Sistema Ficheros
(Sen deben contestar correctamente todas las cuestiones de cada pregunta para puntuar la pregunta).

P1) Un sistema de archivos tipo UNIX tiene un tamaño de bloque de 8Kb, i-nodos con 12 direcciones directas, una indirecta simple, una indirecta doble y una indirecta triple. Utiliza direcciones de bloque de 8 bytes. Calcular cuántos bloques de disco son necesarios para representar el archivo *"datos"* cuando su tamaño es de 8.0234375 Gbytes (8 Gbytes + 24 Mbytes). Discriminar cuántos bloques son de datos y cuántos de índices.

Número de bloques de datos: 1 Mbloques + 3Kbloques

Número de bloques de índices: 1030

P2) Un proceso abre el archivo *"datos"*, *open ("datos", O_RDONLY)*, que se encuentra en el directorio de trabajo (la apertura no retorna ningún error), y lo siguiente que hace el proceso es:

lseek(fd, 7516192768, SEEK_SET); *(SEEK_SET referencia desde el principio de fichero)*

Calcular cuántos bloques tendría el SO que leer del disco, discriminando entre bloques de datos y de índices, para realizar a continuación la operación:
c=fgetc(fd)
suponiendo el buffer caché vacío. (Nota: $7516192768 = 7*2^{30}$)

Número de bloques de índices que es necesario leer: 2

Número de bloques de datos que es necesario leer: 1

P3) Un proceso abre (sin retorno de ningún error) el fichero *“datos”* (descriptor *fd1*) y, posteriormente, duplica el descriptor de fichero abierto con el servicio *dup*, al ejecutar el siguiente código:

```
main()
{
int fd1, fd2;
char buffer[2048];

fd1=open(“datos", O_RDONLY);
read (fd1, buffer, 1024);
lseek(fd1, 900, SEEK_SET); /* SEEK_SET referencia desde el principio del fichero */
fd2=dup(fd1);
read (fd2, buffer, 1024);
lseek(fd2, 700, SEEK_SET);
close(fd2);
}
```

Indicar qué es cierto después de ejecutarse la segunda llamada *lseek()*.

A. El desplazamiento asociado al descriptor *fd1* queda en la posición 1600.

Falso

B. El desplazamiento asociado al descriptor *fd2* queda en la posición 1724.

Falso

C. El desplazamiento asociado al descriptor *fd1* queda en la posición 700, y el desplazamiento asociado al descriptor *fd2* queda en la posición 700.

Cierto

D. El desplazamiento asociado al descriptor *fd1* queda en la posición 900, y el desplazamiento asociado al descriptor *fd2* queda en la posición 700.

Falso

E. Las dos lecturas con las llamadas *read* leen la misma información del fichero (mismo rango de bytes).

Falso

P4) Supongamos la siguiente secuencia de comandos:

1. Se crea un *soft link* al fichero *"datos"* (cuyo número de inodo es 22342, tamaño 8.0234375 Gbytes y num. de *hard links*=3) en su mismo directorio:

ln -s datos slink / el comando ls -l mostraría slink -> datos */*

2. Posteriormente se crea un *hard link* al fichero *slink*:

ln slink slink2 / crea hard link slink2 */*

Contestar a lo siguiente:

A. Indicar el tamaño (Gbytes/Mbytes/bytes) del fichero *slink2*: 5 bytes

B. Indicar cuál es el número de *(hard) links* del fichero *slink2*: 2

C. Una vez creados los ficheros *slink* y *slink2*, al borrar el fichero datos (*rm datos*) se decrementa su número de *hard links*. ¿Se puede acceder al contenido del fichero con número de inodo 22342 a través del link simbólico *slink2?*

No

Misma pregunta a través del fichero *slink*:

No

P5) En un sistema de ficheros FFS (BSD):

A. La estructuración del disco en grupos de cilindros minimiza el tiempo de lectura promedio de los datos

   Cierto

B. La división entre bloques y fragmentos de bloque permite reducir la fragmentación interna del disco

   Cierto

C. Al utilizar *Buffer Cache* se reduce la fragmentación interna del disco

   Falso

SISTEMAS OPERATIVOS I

Segundo curso Grado en Informática. Enero 2016

APELLIDOS, Nombre:_____

1	2	3	T
0.5	0.5	0.75	1.75

1. En un sistema system V R4, el comando ' `priocntl -s -c RT -p` *prio pid* ' pone al proceso de identificador *pid* en la clase en tiempo real con prioridad en tiempor real *pri* (las prioridades en tiempo real van de 0 a 59 que corresponden a prioridades de 100 a 159). Si la salida del comando `ps` es

```
abyecto:~# ps
  PID TTY          TIME CMD
 3926 pts/2    00:00:00 bash
 3932 pts/2    00:00:00 ps
abyecto:~#
```

   ¿que ocurre si el *root* ejecuta el comando `priocntl -s -c RT -p 58 3926`? (JUSTIFÍQUESE LA RESPUESTA)

   a) producirá un error de *permission denied*, *bus error* o similar
   b) dejará la máquina inutilizable
   c) dejará la máquina inutilizable si es un sistema con un solo procesador
   d) dejará la máquina inutilizable salvo que haya algun otro porceso de la clase en tiempo real con una prioridad de 59 (159)
   e) no puede saberse
   f) ninguna de las anteriores

   La respuesta correcta es la f) ninguna de las anteriores. Se pone una prioridad en tiempo real (estática y mayor que las prioridades normales de los procesos) al proceso de pid 3926; sin embargo no ocurre nada puesto que el proceso con *pid* 3926 es un shell, que se pasa todo el rato en espera. Si el proceso con pid 3926 fuese un bucle infinito dejaría un sistema con un solo procesador inutilizable

2. Considérese el siguiente código en C

```
main (int argc, char * argv[])
{
  long long int veces=-1;

  while (--veces);
}
```

   Compilamos dicho código y creamos un proceso que lo ejecuta. Dicho proceso se ejecuta (JUSTIFÍQUESE LA RESPUESTA)

a) todo el tiempo en modo usuario

b) todo el tiempo en modo kernel

c) en modo usuario o modo kernel dependiendo del usuario que cree el proceso

d) en modo usuario o modo kernel dependiendo de la prioridad que se le asigne

e) en modo usuario o modo kernel dependiendo de como se haya creado el proceso

f) no puede saberse

g) ninguna de las anteriores

La respuesta correcta es g) ninguna de las anteriores. Dicho proceso es ejecutará en modo kernel la parte que le corresponde de la llamada *fork()* de su creación, la llamada *exec* de reemplazar su código por el que se ve en el enunciado (hasta ese momento es una copia de su proceso padre) la llamada *exit* que ejecuta al terminar (no se trata de un blucle infinito), y las interrupciones que lleguen durante su ejecución (y también, claro está, su parte correspondiente de los cambios de contexto que se produzcan durante su ejecución). El resto del tiempo, es decir, el bucle, se ejecutará en modo usuario.

3. En un sistema tenemos 4 procesos A, B, C y D cuyas ráfagas duran 7, 3, 1 y 4 y que llegan en los instantes 0, 1, 2 y 3 respecticamente. Plánifiquense con los algoritmos FCFS SJF, SRT y RR de quanto 2.

FCFS	A	A	A	A	A	A	A	B	B	B	C	D	D	D	D
RR	A	A	B	B	C	A	A	D	D	B	A	A	D	D	A
RR(*)	A	A	B	B	A	A	C	D	D	B	A	A	D	D	A
SJF	A	A	A	A	A	A	A	C	B	B	B	D	D	D	D
SRTF	A	B	C	B	B	D	D	D	D	A	A	A	A	A	A

Calcúlense los tiempos de retorno y de espera promedio para cada algorimo y rellénese el siguiente cuadro

	FCFS	**RR**	**SJF**	**SRTF**	**RR(*)**
Tiempo retorno A	7	15	7	15	15
Tiempo retorno B	9	9	10	4	9
Tiempo retorno C	9	3	6	1	5
Tiempo retorno D	12	11	12	6	11
T_r promedio	9.25	9.5	8.75	6.5	10
Tiempo espera A	0	8	0	8	8
Tiempo espera B	6	6	7	1	6
Tiempo espera C	8	2	5	0	4
Tiempo espera D	8	7	8	2	7
T_e promedio	5.5	5.75	5	2.75	6.25

(*) Aqui se ha supuesto que cuando un proceso sale de la CPU en el mismo instante en que llega otro, el que abandona la CPU se coloca ANTES en la cola que el que acaba de llegar.

Apellidos:_____Nombre:_____ DNI: _____

Sistemas Operativos – Grado Ingeniera Informática UDC. Enero 2016
Sólo puede usar lápiz, bolígrafo y calculadora.
Tiempo máximo para todo el examen: 3h

Parte Memoria.

En las preguntas 1, 3 y 4 tienen que ser correctos los resultados finales y los cálculos y/o razonamientos.

1. (0.3 ptos) Considere una arquitectura de memoria con un TLB con tiempo de acceso de 1ns (tanto para un fallo como un acierto en el TLB), tablas de páginas en 4 niveles con 512 entradas cada una. El tiempo de acceso a memoria es 100ns. Suponga que no hay fallos de página a disco. ¿Qué porcentaje de acierto en el TLB se necesita para tener un tiempo efectivo de acceso a memoria de 105 ns?: _____99%_____

   Sea HR (hit rate) la tasa de acierto en el TLB
   HR x 101 + (1- HR) x 501 = 105
   396 = 400 HR
   HR = 0,99

2. (0.25 ptos) Considere el esquema de gestión de memoria con registros base y límite. De las siguientes marque con X las que sean ciertas con esta solución a la gestión de la memoria (puede haber cero, una o varias correctas). Si marca como correcta una incorrecta, se valorará con cero. Para alcanzar la puntuación máxima debe marcar sólo y todas las correctas (o ninguna si fuese el caso).

   __ No proporciona relocalización dinámica
   __ No proporciona protección de la memoria
   X El espacio lógico de direcciones de un proceso es contiguo
   X Tiene el problema de fragmentación externa
   X Un proceso sólo puede tener un segmento

3. (0.6 puntos) Considere la siguiente cadena de referencias a páginas: 1,2,3,4,2,5,7,2,3,2,1,7,8. Para cada uno de los algoritmos de reemplazo siguiente considerando 4 marcos de memoria asignados (páginas físicas), indique el número de fallos de página total, considerando todos los fallos de página aunque no impliquen reemplazo. Muestre también una tabla con la asignación de páginas a frames (marcos).

   FIFO: ____10 fallos____
   LRU:___9 fallos ___
   Optimo: ___7 fallos___

4. a) (0.2 ptos) Considere un sistema con una memoria física de 8 Gigabytes, tamaño de página de 8 kilobytes y entradas en la tabla de página de 4 bytes cada entrada. Con un esquema de gestión de la memoria de tablas de páginas en varios niveles, ¿cuántos niveles son necesarios para trasladar direcciones virtuales de 46 bits si cada tabla de páginas se ajusta a

una página?: ____3____

1 tabla de páginas tiene 8kb/4bytes = 2^{13} / 2^{2} = 2^{11} entradas
por tanto 1 tabla de páginas puede direcciona 2^{11} x 8 kb = 2^{11} x 2^{13} = 2^{24} bytes
con tablas de páginas en dos niveles se podrán direccionar:

2^{11} x 2^{11} x 2^{13} = 2^{35} bytes

con tablas de páginas en tres niveles se podrán direccionar:
2^{11} x 2^{11} x 2^{11} x 2^{13} = 2^{46} bytes

b) (0.2 puntos) En cada entrada de tabla de página ¿que número máximo de bits se pueden usar para información de control (bit de validez, de modificación, r/w/x, etc)? ___12 bits___

8 Gbytes = 2^{33} bytes = 2^{20} páginas de 8Kb = 2^{20} x 2^{13} bytes
Por tanto se necesitan 20 bits en cada entrada de la tabla de páginas para indicar el número de página física, y quedan 12 bits para la otra información

c) (0.2 ptos) ¿Cuanta memoria física necesita como mínimo un proceso que tienen 3 páginas virtuales lógicas (y supuesto todo el proceso en memoria)?: _____6 páginas de 8Kb = 48 Kb _____

Una página con la tabla de páginas de primer nivel, otra página con una tabla de páginas de segundo nivel y una página con una tabla de páginas de tercer nivel. Los «huecos» en el espacio de direcciones virtuales no necesitan entradas en las tablas de páginas. Y a eso hay que sumar las tres páginas del proceso. Si las páginas estuviesen al principio (por ejemplo código y datos) y final del espacio de direcciones lógicos (por ejemplo pila) se necesitarían al menos dos páginas en el segundo nivel y dos en el tercero lo que daría un total de 8 páginas o 64 kb, pero se pregunta por el mínimo.