Apellidos:_____ **Nombre:**_____ **DNI:** _____

Sistemas Operativos – Grado Ingeniería Informática UDC. Enero 2021
Sólo puede usar lápiz, bolígrafo y calculadora.

Parte Sistema Ficheros
Puntuación preguntas: P1: 0.6, P2: 0.5, P3: 0.3, P4:0.6

P1) Un sistema de archivos tipo UNIX System V tiene un tamaño de bloque de 4Kbytes, i-nodos con 10 direcciones directas, una indirecta simple, una indirecta doble y una indirecta triple. Utiliza direcciones de bloque de 8 bytes. Calcular cuántos bloques son necesarios en el área de datos para representar un fichero con un tamaño de 1 Gbyte + 6 Mbytes + 40 Kbytes, diferenciando entre bloques de datos y bloques de índices.

Un sistema de ficheiros tipo UNIX System V ten un tamaño de bloque de 4Kbytes, i-nodos con 10 enderezos directos, un indirecto simple, un indirecto dobre e un indirecto triple. Usa enderezos de bloque de 8 bytes. Calcula cantos bloques son necesarios na área de datos para representar un ficheiro cun tamaño de 1 Gbyte + 6 Mbytes + 40 Kbytes, diferenciando entre bloques de datos e bloques de índices.

A UNIX file system, System V type, has a block size of 4Kbytes, i-nodes with 10 direct addresses, a single indirect, a double indirect and a triple indirect address. It uses 8-byte block addresses. Calculate how many blocks are needed in the data area to represent a file with a size of 1 Gbyte + 6 Mbytes + 40Kbytes, differentiating between data blocks and index blocks.

Nº bloques de datos / Nº bloques de datos / Number of data blocks: 256 Kbloques + 1.5 Kbloques + 10 bloques = 263690 bloques (0.30p)

Nº de bloques de índices / Nº de bloques de índices / Number of index blocks: 518 (0.25p)

Fragmentación interna del fichero/Fragmentación interna do ficheiro/ Internal file fragmentation: 0 (bytes) (0.05p)

Apellidos:_____ **Nombre:**_____ **DNI:** _____

P2) Un proceso abre el archivo anterior (en P1) *"/home/usr1/dir1/datos"* (tamaño 1 Gbyte + 6 Mbytes + 40Kbytes*)* en modo lectura, cuyo número de *hard links* es 2. El usuario efectivo del proceso coincide con el propietario del fichero, y tiene además los permisos de acceso a los directorios *home*, *usr1* y *dir1*. Las cachés de datos e inodos están inicialmente vacías y la entrada *usr1* está en el séptimo bloque del directorio *home*, mientras las demás entradas están en el primer bloque de sus directorios padre. Indicar lo siguiente referente al trozo de código:

Un proceso abre o ficheiro anterior (en P1) *"/home/usr1/dir1/datos"* (tamaño 1 Gbyte + 6 Mbytes + 40Kbytes*)* en modo lectura, cuxo número de *hard links* é 2. O usuario efectivo do proceso coincide co propietario do ficheiro e tamén ten os permisos de acceso os directorios *home*, *usr1* e *dir1*. Os cachés de datos e inodos están inicialmente baleiros e a entrada *usr1* está no séptimo bloque do directorio *home*, mentres que as outras entradas están no primeiro bloque dos seus directorios pai. Indique o seguinte sobre o anaco de código:

A process opens the previous file (in P1) *"/home/usr1/dir1/datos"* (size 1 Gbyte + 6 Mbytes + 40 Kbytes*)* in read mode, whose number of *hard links* is 2. The effective user of the process coincides with the owner of the file, and also has the access permissions to the directories *home*, *usr1* and *dir1*. The data and inode caches are initially empty and the entry *usr1* is in the seventh block of the directory *home*, while the other entries are in the first block of their parent directories. Indicate the following regarding the piece of code:

```
struct stat buf;
char c;
chmod("/home/usr1/dir1/datos", 0442);
if (lstat ("/home/usr1/dir1/datos", &buf)!=-1){
   printf("%s", convertir_permisos(buf.st_mode)); /* se convierten los permisos a formato rwxrwxrwx y se imprimen*/
                                                  /* se converten os permisos a formato rwxrwxrwx e se imprimen*/
                                                  /* permissions are converted to format rwxrwxrwx and printed*/
    int fd=open("/home/usr1/dir1/datos", O_RDONLY); /* es la primera apertura de fichero del proceso */
                                                    /* é a primeira apertura de ficheiro do proceso */
                                                    /* it is the first file open of the process*/
    lseek(fd, 1.073.741.824, SEEK_SET); /* (33.554.432= 2^25) */
          /* SEEK_SET indica que el desplazamiento se considera a partir del origen del fichero */
          /* SEEK_SET indica que o desprazamiento considérase desde a orixe do ficheiro */
          /* SEEK_SET specifies that the offset is considered from the origin of the file */
   c=fgetc(fd);
   close(fd);
   unlink("/home/usr1/dir1/datos");
   }
```

Cada apartado puntúa 0.1p/ Each question scores 0.1p.

A. ¿Cuál es el valor asignado al descriptor de fichero *fd*?:
   Cal é o valor asignado ao descriptor de ficheiro *fd*?:
   What is the value assigned to the file descriptor *fd*?: 3

B. ¿Cuál es el número de accesos necesarios a disco, únicamente en el área de datos, en la apertura del fichero?:
   Cal é o número de accesos ao disco necesarios, só na área de datos, ao abrir o ficheiro?:
   What is the number of disk accesses required, only in the data area, when opening the file?: 10

C. Indica el número de bloques que el SO necesita leer en disco para obtener el valor de *c* en *fgetc(fd)*:
   Indica o número de bloques que o SO cómpre ler desde o disco para obter o valor de *c* en *fgetc (fd)*:
   Indicate the number of blocks that the OS requires to read from disk to get the value of *c* in *fgetc (fd)*: 3

D. Indica los permisos del fichero que se imprimen en formato *rwxrwxrwx*:
   Indica os permisos de ficheiro que se imprimen en formato *rwxrwxrwx*:
   Indicate the file permissions that are printed in *rwxrwxrwx* format: r-- r-- -w-

E. ¿Cuál es el número de hard links de "datos" después de ejecutar *unlink*?:
   Cal é o número de ligazóns duros (*hard links*) de "datos" despois de executar *unlink*?:
   What is the number of hard links of "data" after running *unlink*?: 1

Apellidos:_____ Nombre:_____ DNI: _____

P3) Se crean 3 *hard link* a un fichero "*f1*" (cuyo número de inodo es 25634, tamaño 965 bytes y número de *hard links*=1) en su mismo directorio:
>ln f1 f2
>ln f1 f3
>ln f1 f4
Posteriormente se crea un *soft link* al fichero *f3*:
> ln -s f3 slink /*crea soft link *slink* */
/* el comando *ls -l* mostraría *slink -> f3* */
¿Cuál es el tamaño (en bytes) del fichero "*f3*"?: 965 (bytes) (0.15p)
Finalmente, se borra (comando *rm*) el fichero *f2*
>rm f2
¿Cuál es ahora el tamaño (en bytes) del fichero "*slink*"?: 2 (bytes) (0.15p)

Créanse 3 *hard link* a un ficheiro "*f1*" (cuxo número de inodo é 25634, tamaño 965 bytes e número de *hard links*=1) no mesmo directorio:
> ln f1 f2
> ln f1 f3
> ln f1 f4
Máis tarde, crease un soft link ao ficheiro f3:
*> ln -s f3 slink / *crear soft link slink * /*
*/ * o comando ls -l* amosaría *slink -> f3 * /*
¿Cal é o tamaño (en bytes) do ficheiro "f3"?: 965 (bytes) (0.15p)
Finalmente, se borra (comando *rm*) o ficheiro *f2*
>rm f2
Cal é agora o tamaño (en bytes) do ficheiro "*slink*"?: 2 (bytes) (0.15p)

3 *hard links* are created to a file "*f1*" (whose inode number is 25634, size 965 bytes and number of *hard links*=1) in its same directory:
> ln f1 f2
> ln f1 f3
> ln f1 f4
Next, a soft link to the f3 file is created:
> ln -s f3 slink /* create soft link *slink * /*
/ * the *ls -l* command would show *slink -> f3 * /*
What is the size (in bytes) of the file "*f3*"?: 965 (bytes) (0.15p)
Finally, file *f2* (command *rm*) is removed
>rm f2
What is the size (in bytes) of the file "*slink*" now?: 2 (bytes) (0.15p)

Apellidos:_____ Nombre:_____ DNI: _____

P4) Indicar si es cierto/falso en cada pregunta.

Cada apartado puntúa 0.075p. Cada respuesta errónea puntúa -0.075p. Cuestiones no respondidas no puntúan. La puntuación mínima de P4 es 0, es decir, en ningún caso P4 lleva a puntuación negativa para el total del examen.

A. Mientras un fichero está abierto, el S.O. mantiene una copia de su inodo en memoria principal. C

B. El contenido de un directorio (entradas de directorio) se almacena en el área de datos. C

C. El código lincado con una librería dinámica es “autocontenido”. F

D. El código binario de *fork()* se encuentra en la librería estándar de C (*libC*). F

E. Es posible crear *hard links* entre diferentes sistemas de ficheros montados. F

F. El tipo de fichero “link simbólico” se codifica en el campo “modo” del inodo (el mismo campo en el que se codifican también los permisos del fichero). C

G. La idea fundamental de un sistema de ficheros Unix basado en registro (*journaling file system*) es llevar control de todos los ficheros que se crean y eliminan. F

H. El *Buffer Cache* minimiza posibles errores, ante caídas de alimentación, cuando se realizan operaciones de cambio de datos/metadatos en el sistema de ficheros. F

Indique se é verdadeiro/falso en cada pregunta.

Cada apartado puntúa 0.075p. Cada resposta incorrecta puntúa -0.075p. Cuestións non respondidas non puntúan. A puntuación mínima para P4 é 0, é dicir, en ningún caso P4 ten puntuación negativa para o exame total.

A. Mentres un ficheiro está aberto, o S.O. garda unha copia do seu inodo na memoria principal. C

B. O contido dun directorio (entradas de directorio) almacénase na área de datos. C

C. O código *lincado* cunha biblioteca dinámica é “autocontido”. F

D. O código binario de *fork()* atópase na librería estándar de C (*libC*). F

E. É posible crear vínculos duros (*hard links*) entre diferentes sistemas de ficheiros montados. F

F. O tipo de ficheiro “ligazón simbólica” (*symbolic link*) está codificado no campo “modo” do inodo (o mesmo campo no que tamén están codificados os permisos do ficheiro). C

G. A idea fundamental dun sistema de ficheiros Unix basado en rexistro (*journaling file system*) é facer un seguimento de todos os ficheiros creados e eliminados. F

H. A *Buffer Cache* minimiza os posibles erros, en caso de caída de enerxía, cando se realizan operacións de cambio de datos/metadatos no sistema de ficheiros. F

Indicate whether it is true/false for each question.

Each question scores 0.075p. Each wrong answer scores -0.075p. Unanswered questions do not score.The minimum score for P4 is 0, that is, in no case does P4 have a negative score for the total exam.

A. While a file is open, the O.S. keeps a copy of its inode in main memory. T

B. The content of a directory (directory entries) is stored in the data area. T

C. The code linked with a dynamic library is “self-contained”. F

D. The binary code of *fork ()* is in the standard C library (*libC*). F

E. It is possible to create *hard links* between different mounted file systems. F

F. The type of file “symbolic link” is encoded in the field "mode" of the inode (the same field in which the file permissions are also encoded). C

G. The fundamental idea of a Unix j*ournaling file system* is to keep track of all the files that are created and deleted. F

H. The *Buffer Cache* minimizes possible errors, in the event of power failure, when data/metadata change operations are performed in the file system. F

Apellidos:_____Nombre:_____ DNI: _____

Sistemas Operativos - Grado Ingeniera Informática UDC. Enero 2021
Parte Memoria (2 puntos)

1. Cada una de las preguntas siguientes son de responder V/F (Verdadero/Falso) o no responder. Respuesta correcta: +0.1; Respuesta incorrecta: -0.1. (0.1 x 10 = 1 punto) Puntuación mínima para esta pregunta: 0.
Cada unha das preguntas seguintes son de responder V/F (Verdadeiro/Falso) ou non responder. Resposta correcta: +0.1; Resposta incorrecta: -0.1. (0.1 x 10 = 1 punto) Puntuación mínima para esta pregunta: 0.
Answer T/F (True/False) or leave the answer blank. Correct answer: 0.1; Wrong answer: -0.1 (0.1 x 10 = 1 point) Minimum score for this question: 0.

- **En la pila del proceso está el código de las funciones recursivas cuando se llaman**: __F___
Na pila do proceso está o código das funcións recursivas cando se chaman
The process stack contains the code of recursive functions when they are called

-Cuando se ejecuta un programa C, las variables locales del main() están en la pila del proceso: _V____
Cando se executa un programa C, as variables locais de main() están na pila do proceso
When running a C program, the local variables of main() are in the process stack

-Para un proceso con varios hilos (threads) de ejecución, los hilos comparten código, datos y pila: __F__
Para un proceso varios fíos de execución (threads), os fíos comparten código, datos e pila.
For a process consisting of several threads, the threads share code, data and stack.

-El código de la función de librería malloc usada en un programa C está en el espacio de direcciones del proceso:__V__
O código da función de librería malloc usado nun programa C está no espacio de direccións do proceso
The code for the malloc library function used in a C program is in the process address space

-Para un sistema con una tabla de páginas de tres niveles (sin caché, ni TLB), traer y ejecutar una instrucción que añade un valor constante a un registro, implica exactamente tres accesos a memoria: _F__
Para un sistema con táboa de páxinas en tres niveis (sin caché, nin TLB), traer e executar unha instrucción que engade un valor constante a un rexistro, implica exactamente tres accesos a memoria.
For a system with 3 levels page table (without caché, without TLB), fetch and execute an instruction with adds a constant value to a register, implies exactly three memory accesses.

-Para un sistema con tablas de páginas en tres niveles, donde la tabla de páginas raíz no está fija en memoria, y procesador con caché y TLB, traer y ejecutar una instrucción que añade un valor constante a un registro, puede implicar cuatro accesos a memoria: _V__
Para un sistema con táboa de páxinas en tres niveis, donde a táboa de páxinas non está fixa en memoria, e procesador con caché e TLB, traer e executar unha instrucción que engade un valor constante a un rexistro, pode implicar catro accesos a memoria.
For a system with three-level page table, the root page table is not locked in memory, and processor with caché and TLB, fetching and executing an instruction that adds a constant value to a register, may involve four memory accesses.

-Si las direcciones físicas son de 32 bits y las páginas de 4Kbytes, el número de página virtual necesariamente viene dado por 20 bits:_F__
Si as direccións físicas son de 32 bits e as páxinas de 4Kbytes, o número de páxina virtual necesariamente ven dado por 20 bits.
With 32 bits physical addresses and 4Kbytes pages, it is always the case of 20 bits virtual page numbers.

-En un sistema con segmentación que NO TIENE paginación (segmentación pura), es posible implementar memoria virtual:_V_
En un sistema con segmentación que NON TEN paxinación (segmentación pura), é posible implementar memoria virtual
For a system with segmentation and no paging (pure segmentation), it is possible to implement virtual memory

-Con Tabla de Páginas Invertida, los sistema operativos no podrían resolver los fallos de páginas de los procesos:__F__
Con Táboa de Páxinas Invertida, os sistemas operativos non podrían resolver os fallos de páxina dos procesos
With the technique of Inverted Page Table, operating systems could not resolve process page faults

-**Para un proceso cuyo espacio virtual ocupa N páginas, el algoritmo de reemplazo FIFO produce siempre los mismos fallos de página con N marcos asignados al proceso que con N+1:**__V__
Para un proceso con un espacio virtual de N páxinas, o algoritmo de reemplazo FIFO produce sempre os mesmos fallos de páxina con N marcos asignados ó proceso que con N+1
For a process with a N pages virtual space, the FIFO replacement algorithms shows always the same number of page faults with N frames assigned to the process than with N+1 frames assigned

2. (0.4 puntos)
2.v1.**Un proceso tiene la cadena de referencias a páginas que se muestra y tiene asignados tres marcos de memoria. Las tres primeras referencias, estos es, la referencias a las páginas E, D, H, producen necesariamente 3 fallos de página porque ninguna página del proceso estaba en memoria. ¿Cual es el número total de fallos de páginas que se producen con el algoritmo de reemplazo LRU (Least Recently Used)? Obviamente hay que contar esos 3 fallos iniciales en el total. Tanto la asignación de páginas a frames como el total de número de fallos deben ser correctos para puntuar la pregunta.**
Un proceso ten a cadea de referencias a páxinas que se mostra e ten asignadas tres marcos de memoria. As tres primeiras referencias, isto é, as referencias as páxinas E, D, H, producen necesariamente 3 fallos de páxina porque ningunha páxina do proceso estaba en memoria. ¿Cal é o número total de fallos de páxina que se producen con algoritmo de reemplazo LRU (Least Recently Used)? Obviamente hay que contar esos 3 fallos iniciais no total. Tanto a asignación de páxinas a frames como o total de número de fallos deben ser correctos para puntuar a pregunta.
The string of page references for a process that has been allocated 3 page frames is the one shown below. The first three page references, i.e. references to pages E, D, H, produce 3 page faults because none of the pages of this process were in memory. What is the total number of page faults applying the LRU (Least Recently Used) replacement algorithm? Obviously, you have to include the initial three faults in the total amount. Both the assignment of pages to frames and the total number of page faults must be right to get the score for this question.
Cadena de referencias. String of page references:
E D H B D E D A E B E D E B G **Respuesta: 9 fallos**

E	D	H	B	D	E	D	A	E	B	E	D	E	B	G
E	E	E	B	B	B	B	A	A	A	A	D	D	D	G
	D	D	D	*	D	*	D	D	B	B	B	B	*	B
		H	H	H	E	E	E	*	E	*	E	*	E	E
F	F	F	F		F		F		F		F			F

2.v2. **Un proceso tiene la cadena de referencias a páginas que se muestra y tiene asignados tres marcos de memoria. Las tres primeras referencias, estos es, la referencias a las páginas E, D, H, producen necesariamente 3 fallos de página porque ninguna página del proceso estaba en memoria. ¿Cual es el número total de fallos de páginas que se producen con el algoritmo de reemplazo LRU (Least Recently Used)? Obviamente hay que contar esos 3 fallos iniciales en el total. Tanto la asignación de páginas a frames como el total de número de fallos deben ser correctos para puntuar la pregunta.**
Un proceso ten a cadea de referencias a páxinas que se mostra e ten asignadas tres marcos de memoria. As tres primeiras referencias, isto é, as referencias as páxinas E, D, H, producen necesariamente 3 fallos de páxina porque ningunha páxina do proceso estaba en memoria. ¿Cal é o número total de fallos de páxina que se producen con algoritmo de reemplazo LRU (Least Recently Used)? Obviamente hay que contar esos 3 fallos iniciais no total. Tanto a asignación de páxinas a frames como o total de número de fallos deben ser correctos para puntuar a pregunta.
The string of page references for a process that has been allocated 3 page frames is the one shown below. The first three page references, i.e. references to pages E, D, H, produce 3 page faults because none of the pages of this process were in memory. What is the total number of page faults applying the LRU (Least Recently Used) replacement algorithm? Obviously, you have to include the initial three faults in the total amount. Both the assignment of pages to frames and the total number of page faults must be right to get the score for this question.
Cadena de referencias. String of page references:
E D H B D E D A E B E D E B D **Respuesta: 8 fallos**

E	D	H	B	D	E	D	A	E	B	E	D	E	B	D
E	E	E	B	B	B	B	A	A	A	A	D	D	D	*
	D	D	D	*	D	*	D	D	B	B	B	B	*	B
		H	H	H	E	E	E	*	E	*	E	*	E	E
F	F	F	F		F		F		F		F			

3. (0.6 puntos: 0.3 cada apartado A) B))

3.v1. **Una arquitectura de memoria tiene direcciones virtuales de 64 bits y direcciones físicas de 52 bits. Hay un número de bits no usados en la dirección virtual. Tiene páginas de 256 Kbytes (1 Kbytes= 1024 bytes), tabla de páginas en dos niveles con el mismo número de bits en las direcciones virtuales para cada uno de los dos niveles y entradas en la tabla de páginas de 8 bytes. A) ¿Cuantos bits NO usados hay en una dirección virtual? B) ¿Cuántos bits se**

usan en una dirección virtual para seleccionar la entrada de la tabla de páginas de raíz? Tanto los cálculos como el resultado final tienen que ser correctos para obtener la puntuación
Unha arquitectura de memoria ten direccións virtuais de 64 bits e direccións físicas de 52 bits. Hay un número de bits non usados nas direccións virtuais. Ten páxinas de 256 Kbytes (1 Kbytes= 1024 bytes), táboa de páxinas en dous niveis co mismo número de bits nas direccións virtuais para cada un dous niveis e entradas na táboa de páxinas de 8 bytes. A) ¿Cántos bits NON usados hay nunha dirección virtual? B)¿Cantos bits se usan nunha dirección virtual para seleccionar a entrada da táboa de páxinas raíz? Tanto os cálculos como o resultado final teñen que ser correctos para obter a puntuación
Consider a memory architecture with a 64 bit virtual address space and a 52 bit physical address space, 256 Kbytes page size (1 Kbytes= 1024 bytes) and a two-level page table with the same number of virtual bits for both levels. Each page table entry is 8 bytes. A) How many NOT used bits does a virtual address have? B) How many bits are used in a virtual address to select the root page entry? Both the calculations and final result must be right to get the score for this question
A) 16 bits.
B) 15 bits.

256Kbytes= 2^{18} bytes, (2^{18} bytes/pag) / (2^{3} bytes/entrada) = 2^{15} entradas en cada TP
formato entrada TP:
16 bits no usados, 15 bits para seleccionar entrada TP 1er nivel, 15 bits para seleccionar entrada TP 2nd nivel, 18 bits offset

3.v2 **Una arquitectura de memoria tiene direcciones virtuales de 64 bits y direcciones físicas de 52 bits. Hay un número de bits no usados en la dirección virtual. Tiene páginas de 128 Kbytes (1 Kbytes= 1024 bytes), tabla de páginas en dos niveles con el mismo número de bits en las direcciones virtuales para cada uno de los dos niveles y entradas en la tabla de páginas de 8 bytes. A) ¿Cuantos bits NO usados hay en una dirección virtual? B) ¿Cuántos bits se usan en una dirección virtual para seleccionar la entrada de la tabla de páginas de raíz? Tanto los cálculos como el resultado final tienen que ser correctos para obtener la puntuación**
Unha arquitectura de memoria ten direccións virtuais de 64 bits e direccións físicas de 52 bits. Hay un número de bits non usados nas direccións virtuais. Ten páxinas de 128 Kbytes (1 Kbytes= 1024 bytes), táboa de páxinas en dous niveis con mismo número de bits nas direccións virtuais para cada un dous niveis e entradas na táboa de páxinas de 8 bytes. A) ¿Cántos bits NON usados hay nunha dirección virtual? B)¿Cantos bits se usan nunha dirección virtual para seleccionar a entrada da táboa de páxinas raíz? Tanto os cálculos como o resultado final teñen que ser correctos para obter a puntuación
Consider a memory architecture with a 64 bit virtual address space and a 52 bit physical address space, 128 Kbytes page size (1 Kbytes= 1024 bytes) and a two-level page table with the same number of virtual bits for both levels. Each page table entry is 8 bytes. A) How many NOT used bits does a virtual address have? B) How many bits are used in a virtual address to select the root page entry? Both the calculations and final result must be right to get the score for this question
A) 19 bits.
B) 14 bits.

128Kbytes= 2^{17} bytes, (2^{17} bytes/pag) / (2^{3} bytes/entrada) = 2^{14} entradas en cada TP
formato entrada TP:
19 bits no usados, 14 bits para seleccionar entrada TP 1er nivel, 14 bits para seleccionar entrada TP 2nd nivel, 17 bits offset

Apellidos:_____ **Nombre:**_____ **DNI:** _____

Sistemas Operativos – Grado Ingeniería Informática UDC. Enero 2021
Examen SO. Procesos

1.-**Cada una de las preguntas siguientes son de responder V/F (Verdadero/Falso) o no responder. Respuesta correcta: +0.1; Respuesta incorrecta: -0.1. (0.1 x 10 = 1 punto) Puntuación mínima para esta pregunta: 0.**
Cada unha das preguntas seguintes son de responder V/F (Verdadeiro/Falso) ou non responder. Resposta correcta: +0.1; Resposta incorrecta: -0.1. (0.1 x 10 = 1 punto)
Answer T/F (True/False) or leave the answer blank. Correct answer: 0.1; Wrong: -0.1

-La credencial real y la credencial efectiva de un proceso son siempre iguales entre si/*Real credential always matches the efective credential/*A credencial real e a credencial efectiva dun proceso sempre son iguais entre si___**F**

-Las variables de entorno son iguales para todos los procesos del sistema/*Environment variables are the same for all processes in the system/*As variables de entorno son iguais para todos os procesos do sistema___**F**

-Las variables de entorno son iguales para todos los procesos del mismo usuario/*Environment variables are the same for all processes of the same user/*As variables de entorno son iguais para todos os procesos do mesmo usuario___**F**

-Cada proceso tiene su copia de los datos del kernel/*Each process has its copy of the kernel data/*Cada proceso ten a súa copia dos datos do núcleo___**F**

-La pila del kernel debe estar protegida de accessos concurrentes/*The kernel stack must be protected from concurrent accesses/*A pila do núcleo debe estar protexida de accesos simultáneos___ **F**

-La transición de ejecución a espera es SIEMPRE desde ejecución en modo kernel/*The transition from running to blocked is ALWAYS from kernel running/*A transición de executar a esperar SEMPRE é dende executar en modo núcleo___**V**

-El estado de apropiado (preempted) es el mismo que el de listo para ejecución/*The preempted state is the same as runnable (ready to run) state/*O estado apropiado (preempted) é o mesmo que listo para executar___**V**

-Cuando un programa comienza su ejecución las variables de entorno están en la pila de usuario/ *When a program starts execution, environment variables are stored in the user stack/*Cando un programa comeza a súa execución, as variables de entorno están na pila de usuario___**V**

-La llamada execvp() crea un proceso que ejecuta un prgrama que está en el PATH/*The execvp() system call creates a process that executes a program which is in the PATH*___
-A chamada execvp() crea un proceso que executa un programa que está no PATH___**F**

-Suponiendo que se llama desde el proceso padre del proceso pi2, waitpid(pi2,NULL,0) desasigna la estructura proc del proceso pi2 cuando termina/*Assuming the calling process is pi2's parent process, waitpid(pi2,NULL,0) deallocates pi2's proc struct/*Supoñendo que se chama desde o proceso pai do proceso pi2 ,waitpid (pi2, NULL, 0) deslocala a estrutura proc do proceso pi2 cando remata___**V**

Apellidos:_____ Nombre:_____ DNI: _____

2.-(0.5 puntos) El siguiente código crea un proceso que ejecuta un programa que se le pása con sus párametros en un array de punteros terminado a NULL. (argv[0] es el nombre del ejecutable, argv[1] el primer parámetro...). Contéstese si es correcto o no y respóndase la cuestión correspondiente

The following code creates a process that executes a program that receives its parameters in a NULL terminaded array (argv[0] is the name of the executable file, argv[1] the first parameter...)

O seguinte código crea un proceso que executa un programa que se pasa cos seus parámetros nun array de punteiros rematado por NULL. (argv [0] é o nome do executable, argv [1] o primeiro parámetro ...)

```
void Execute (char *argv[])
{
   pid_t pid;

   if ((pid=fork())==0){
       if (execvp(argv[0],argv)==-1)
          perror («cannot execute»);
       }
  else
      waitpid (pid,NULL,0);
}
```

Es incorrecto, una vez creado el proceso con fork, si execvp falla, el proceso hijo seguiría existiendo y quedarían dos procesos ejecutándose. Falta poner una llamada a exit despues del perror

```
void Execute (char *argv[])
{
   pid_t pid;

   if ((pid=fork())==0){
       if (execvp(argv[0],argv)==-1)
          perror («cannot execute»);
       exit (0);
       }
  else
      waitpid (pid,NULL,WNOHANG);
}
```

Es correcto, pero dado que waitpid lleva el flag WNOHANG que hace que waitpid no espere, la ejecución es en segundo plano (en primer plano el proceso padre esperaría a que el proceso hijo terminase)

Apellidos:_____ Nombre:_____ DNI: _____

```
void Execute (char *argv[])
{
   pid_t pid;

  if (execvp(argv[0],argv)==-1){
        perror («cannot execute»);
        exit(0);
       }
  else
      waitpid (pid,NULL,0);
}
```

Es incorrecto, execvp NO CREA ningún proceso, falta que haya una llamada a fork para crear un proceso

3.-(0.5 puntos) En un sistema hay dos procesos A y B con duraciones 4-(5)-3 (ráfaga de CPU de 4, seguida de ráfaga de E/S de 5, seguida de ráfaga de CPU de 3) y 2-(4)-1, (ráfaga de CPU de 2, seguida de ráfaga de E/S de 4, seguida de ráfaga de CPU de 1) respectivamente. Los instantes de llegada son 0 para A y 1 para B.

Rellenar la siguiente tabla con el tiempo de retorno para A, tiempo de retorno para B y el porcentaje de uso de la CPU (en %), para los casos de multiprogramación con FCFS, multiprogramación con SRTF y no multiprogramación

In a system there are two processes A and B with durations 4-(5)-3 (CPU burst of 4, followed by I/O burst of 5, followed by CPU burst of 3) and 2-(4)-1, (CPU Burst 2, followed by I/O Burst 4, followed by CPU Burst 3) respectively. The arrival times are 0 for A and 1 for B.

Fill in the following table with the turnaround time for A, turnaround time for B and the perecentage of CPU usage, for the FCFS multiprogramming, SRTF multiprogramming and non-multiprogramming cases

	FCFS	**SRTF**	**NO Multiprogram**
Tiempo Retorno A	**12**	**14**	**12**
Tiempo Retorno B	**12**	**7**	**18**
Uso de CPU (%)	**76.9**	**71.4**	**52.6**

Apellidos:_____**Nombre:**_____ **DNI:** _____

Escriba claramente Apelidos y nombre en mayúsculas / Use capital letters for Surname and name

Sistemas Operativos - Grado Ingeniera Informática UDC. Xaneiro de 2021 Opción A

Parte E/S (1.5 puntos).

1: (0.5 puntos/points)

Respóndase V/F (Verdadero/Falso) o deje en blanco. Respuesta correcta: +0.1; Respuesta incorrecta: -0.1. Puntuación mínima en esta pregunta: 0.
Respóndase V/F (Verdadeiro/Falso) ou deixe en branco. Resposta correcta: +0.1; Resposta incorrecta: -0.1. Puntuación mínima para esta pregunta: 0.
Answer T/F (True/False) or leave as blank. Correct answer: +0.1; Wrong answer: -0.1. Minimum score for this question: 0.

A	B	C	D	E

← Responde V/F aquí : o/ou responde en blanco
either answer T/F here, or leave blank

A- **El registro de control de un controlador de dispositivo permite saber si el dispositivo está disponible para recibir/transmitir un nuevo dato**. __F__
O rexistro de control dun controlador de dispositivo permite saber se o dispositivo está dispoñible para recibir/transmitir un novo dato.
In a device controller, the control register permits us to know if the device is ready to receive/transmit a new piece of data.

B- **Considérese la estructura en capas del software de E/S. Asumamos un disco duro con un sistema de ficheros que usa bloques de 1Kbyte. Cuando carguemos un fichero de 2500 bytes en un editor de textos, la capa denominada software independiente del dispositivo se encarga de calcular cuántos sectores de disco deben ser leídos:** _F___
Considérese a estructura en capas do software de E/S. Asumamos un disco duro cun sistema de ficheiros que usa bloques de 1Kbyte. Cando carguemos un ficheiro de 2500 bytes nun editor de textos, a capa denominada software independente de dispositivo encárgase de calcular cantos sectores de disco deben ser lidos.
Let us consider the layered structure of the I/O software. Lets assume a hard drive containing a file system that uses 1Kbyte blocks. When we load a file of size 2500 bytes into a text editor, the layer named device-independent software is in charge of computing the number of sectors that must be read from disk.

C- **Un disco duro con 4 caras y 16000 pistas por cara, tiene 64000 cilindros:** _F___
Un disco duro con 4 caras e 16000 pistas por cara, ten 64000 cilindros.
A hard disk with 4 sides and 16000 tracks in each side contains 64000 cylinders.

D- **Considerando la siguiente salida del ls: El *major-number* del dispositivo /dev/sdb es 8:** _V___
Considerando a seguinte saída do ls: O major-number do dispositivo /dev/sdb é 8.
Considering the following output of the *ls* command: The major number of the device /dev/sdb is 8.

```
user@beowulf:~$ ls -l /dev/sd*
 brw-rw---- 1 root disk  8,  0 nov 25 12:20 /dev/sda
 brw-rw---- 1 root disk  8,  1 nov 25 12:20 /dev/sda1
 brw-rw---- 1 root disk  8,  2 dic 31 01:25 /dev/sda2
 brw-rw---- 1 root disk  8, 16 nov 25 12:20 /dev/sdb
 brw-rw---- 1 root disk  8, 17 nov 25 12:20 /dev/sdb1
```

E- **Un controlador DMA configurado en modo ráfaga tiene mayor prioridad de acceso al bus que otro configurado en modo robo de ciclos:** _V___
Un controlador DMA configurado en modo ráfaga ten maior prioridade de acceso ao bus que outro configurado en modo roubo de ciclos.
A DMA controller configured in burst mode has a higher priority when accessing the bus than other one configured in cycle-stealing mode.

2: (0.5 puntos/points)

En un determinado instante, la cola de peticiones de E/S para acceder a cilindros de un disco contenía las peticiones: [25, 100, 360, 77, 125, 222]. Ya se han atendido las peticiones [100] (en primer lugar) y [77]. Indique cuál será el siguiente cilindro accedido si utilizamos los algoritmos:

Nun determinado instante, a cola de peticións de E/S para acceder a cilindros dun disco contiña as peticións: [25,100, 360, 77, 125, 222]. Xa se atenderon as peticións [100] (en primeiro lugar) e [77]. Indique cal será o seguinte cilindro accedido se utilizamos os algoritmos:

At a given time, the I/O requests queue for accessing cylinders within a disk contained the following requests [25,100, 360, 77, 125, 222]. Requests [100] (in first place) and [77] have already been attended. Which will be the next accessed cylinder according to the following algorithms: ?

SSTF: _____ 125 Tras 100, 77 --> El más próximo a 77 es 125

C-SCAN: _____ 25 Tras 100,77 --> ascensor que atiende peticiones sólo en sentido descendente --> La siguiente petición será la 25

SCAN: _____ 25 Tras 100,77 --> el ascensor (que este caso atiende peticiones tanto al bajar como al subir), en este caso sabemos que el ascensor está bajando --> la siguiente petición atendida será la 25

3: (0.25 puntos/points)

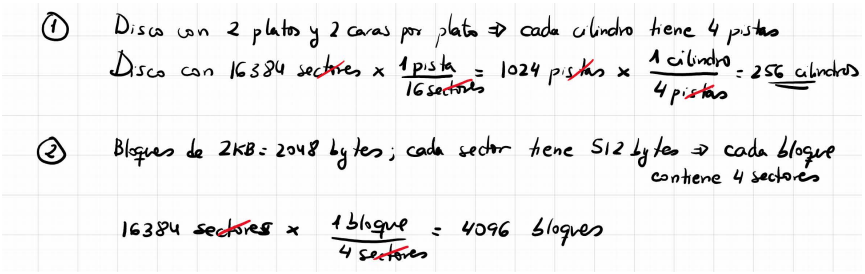
Tenemos un disco que contiene 16384 (=16*1024) sectores. El disco tiene 2 platos y 2 caras en cada plato. Cada sector contiene 512 bytes, y cada pista contiene 16 sectores. ¿Cuántos cilindros tiene? ¿Cuántos bloques verá el S.O. si se formatea usando bloques de 2048 bytes?

*Temos un disco que contén 16384 (=16*1024) sectores. O disco ten 2 platos e 2 caras en cada plato. Cada sector contén 512 bytes, e cada pista contén 16 sectores. ¿Cantos cilindros ten?¿Cantos bloques verá o S.O. se se formatea usando bloques de 2048 bytes?*

We have a disk composed of 16384 (=16*1024) sectors. The disc has 2 platters and there are 2 sides in each platter. Each sector contains 512 bytes, and each track contains 16 sectors. How many cylinders are there on the disk? How many blocks will be seen by the O.S. if it is formated using blocks of 2048 bytes each?

Num cilindros/cylinders: 256

Num bloques/blocks: 4096



4: (0.25 puntos/points)

El fichero "fichero.txt" contiene "01234567890123456789\0\n". ¿Cuál será el contenido de BUF[i] (i=1..4) TRAS ejecutar la instrucción en la línea 08 del siguiente programa?

El fichero "file.txt" contiene "01234567890123456789\0\n". Cuál será el contenido de BUF[i] (i=1..4) TRAS ejecutar la instrucción en la línea 08 del siguiente programa?

The file "file.txt" contains "01234567890123456789\0\n". Which will be the contents of BUF[i] (i=1..4) AFTER executing the intruction at line 08 in the following program?

After/tras lin.02: **BUF[i] =**	-	-	-	-	\0
After/tras lin.08: **BUF[i] =**					
i =	0	1	2	3	4

```
l.01 int main(){
l.02     char BUF[5]= {'-','-','-','-','\0'};
l.03     int fd = open("fichero.txt",O_RDONLY);
l.04     int fd2 = dup(fd);
l.05     lseek(fd2, 2, SEEK_SET);
l.06     pread(fd,BUF,1,0);            0 - - - \0
l.07     lseek(fd, 2, SEEK_CUR);
l.08     pread(fd,BUF+1,3,1);          0 1 2 3 \0
l.09     printf("%s\n",BUF);
l.10     read(fd,BUF,3);
l.11     read(fd,BUF+2,2);
l.12     printf("%s",BUF); close(fd);
l.13   }
```

resposta xa.c => 0 1 2 3 \0

Apellidos:_____**Nombre:**_____ **_ DNI:** _____

Sistemas Operativos - Grado Ingeniera Informática UDC. Xaneiro de 2021 Opción B

Parte E/S (1.5 puntos).

*Escriba **claramente** Apelidos y nombre en mayúsculas / Use capital letters for Surname and name*

1: (0.5 puntos/points)

Respóndase V/F (Verdadero/Falso) o deje en blanco. Respuesta correcta: +0.1; Respuesta incorrecta: -0.1. Puntuación mínima en esta pregunta: 0.
Respóndase V/F (Verdadeiro/Falso) ou deixe en branco. Resposta correcta: +0.1; Resposta incorrecta: -0.1. Puntuación mínima para esta pregunta: 0.
Answer T/F (True/False) or leave as blank. Correct answer: +0.1; Wrong answer: -0.1. Minimum score for this question: 0.

A	B	C	D	E

← Responde V/F aquí : o/ou responde en blanco
either answer T/F here, or leave blank

A- **El registro de datos de un controlador de dispositivo permite saber si el dispositivo está disponible para recibir/transmitir un nuevo dato**: __F___
O rexistro de datos dun controlador de dispositivo permite saber se o dispositivo está dispoñible para recibir/transmitir un novo dato.
In a device controller, the data register permits us to know if the device is ready to receive/transmit a new piece of data.

B- **Considérese la estructura en capas del software de E/S. La capa denominada software independiente de dispositivo es la encargada de chequear si un proceso tiene permisos para abrir un fichero:** _V___
Considérese a estructura en capas do software de E/S. A capa denominada software independente do dispositivo é a encargada de chequear se un proceso ten permiso para abrir un ficheiro.
Let us consider the layered structure of the I/O software. The layer named device independent software is in charge of checking if a process has enough permissions to open a file.

C- **Un disco duro con 4 caras y 16000 cilindros, tiene 64000 pistas:** _V___
Un disco duro con 4 caras e 16000 cilindros, ten 64000 pistas.
A hard disk with 4 sides and 16000 cylinders contains 64000 tracks.

D- **Considerando la siguiente salida del ls: El *major-number* del dispositivo /dev/sda es 0:** _F___
Considerando a seguinte saída do ls: O major-number do dispositivo /dev/sda é 0.
Considering the following output of the *ls* command: The major number of the device /dev/sda is 0.

```
user@beowulf:~$ ls -l /dev/sd*
 brw-rw---- 1 root disk  8,  0 nov 25 12:20 /dev/sda
 brw-rw---- 1 root disk  8,  1 nov 25 12:20 /dev/sda1
 brw-rw---- 1 root disk  8,  2 dic 31 01:25 /dev/sda2
 brw-rw---- 1 root disk  8, 16 nov 25 12:20 /dev/sdb
 brw-rw---- 1 root disk  8, 17 nov 25 12:20 /dev/sdb1
```

E- **Un controlador DMA configurado en modo bus transparente tiene mayor prioridad de acceso al bus que otro configurado en modo robo de ciclos:** _F___
Un controlador DMA configurado en modo bus transparente ten maior prioridade de acceso ao bus que outro configurado en modo roubo de ciclos.
A DMA controller configured in transparent-bus mode has a higher priority when accessing the bus than other one configured in cycle-stealing mode.

2: (0.5 puntos/points)

En un determinado instante, la cola de peticiones de E/S para acceder a cilindros de un disco contenía las peticiones: [35, 100, 360, 77, 125, 222]. Ya se han atendido las peticiones [77] (en primer lugar) y [35]. Indique cuál será el siguiente cilindro accedido si utilizamos los algoritmos:

Nun determinado instante, a cola de peticións de E/S para acceder a cilindros dun disco contiña as peticións: [35, 100, 360, 77, 125, 222]. Xa se atenderon as peticións [77] (en primeiro lugar) e [35]. Indique cal será o seguinte cilindro accedido se utilizamos os algoritmos:

At a given time, the I/O requests queue for accessing cylinders within a disk contained the following requests [35, 100, 360, 77, 125, 222]. Requests [77] (in first place) and [35] have already been attended. Which will be the next accessed cylinder according to the following algorithms: ?

SSTF: _____ 100

Tras atender 77 y después 35, el más próximo sería el 100

C-SCAN: _____ 360

Tras atender 77 y 35 --> ascensor que atiende en sentido "descendente", pero ya no quedan más bajando --> atenderá 360 de nuevo cuando vuelva a bajar

SCAN: _____ 100

Tras atender 77 y 35 el ascensor está bajando y atendiendo peticiones. Como no hay más en sentido descendente, llegará hasta el cilindro 0, desde donde volverá a subir --> la primera petición que se atenderá en sentido "ascendente" será la del cilindro 100

3: (0.25 puntos/points)

Tenemos un disco que contiene 32768 (=32*1024) sectores. El disco tiene 2 platos y 2 caras en cada plato. Cada sector contiene 512 bytes, y cada pista contiene 8 sectores. ¿Cuántos cilindros tiene? ¿Cuántos bloques verá el S.O. si se formatea usando bloques de 8192 bytes?

*Temos un disco que contén 32768 (=32*1024) sectores. O disco ten 2 platos e 2 caras en cada plato. Cada sector contén 512 bytes, e cada pista contén 8 sectores. ¿Cantos cilindros ten?¿Cantos bloques verá o S.O. se se formatea usando bloques de 8192 bytes?*

We have a disk composed of 32768 (=32*1024) sectors. The disc has 2 platters and there are 2 sides in each platter. Each sector contains 512 bytes, and each track contains 8 sectors. How many cylinders are there on the disk? How many blocks will be seen by the O.S. if it is formated using blocks of 8192 bytes each?

Num cilindros/cylinders: 1024

Num bloques/blocks: 2048

① Disco con 2 platos y 2 caras por plato → cada cilindro tiene 4 pistas

Disco con 32768 sectores × $\frac{1\text{ pista}}{8\text{ sectores}}$ = 4096 pistas × $\frac{1\text{ cilindro}}{4\text{ pistas}}$ = 1024 cilindros.

② Bloques de 8192 bytes; cada sector tiene 512 bytes → cada bloque contiene 16 sectores.

32768 sectores × $\frac{1\text{ bloque}}{16\text{ sectores}}$ = 2048 bloques

4: (0.25 puntos/points)

El fichero "fichero.txt" contiene "01234567890123456789O\n". ¿Cuál será el contenido de BUF[i] (i=1..4) TRAS ejecutar la instrucción en la línea 08 del siguiente programa?

El fichero "file.txt" contiene "01234567890123456789O\n". Cuál será el contenido de BUF[i] (i=1..4) TRAS ejecutar la instrucción en la línea 08 del siguiente programa?

The file "file.txt" contains "01234567890123456789O\n". Which will be the contents of BUF[i] (i=1..4) AFTER executing the intruction at line 08 in the following program?

After/tras lin.02: **BUF[i] =**	-	-	-	-	\0
After/tras lin.08: **BUF[i] =**					
i =	0	1	2	3	4

<-- Fill your answer here
<-- Contesta aquí

```
l.01 int main(){
l.02     char BUF[5]= {'-','-','-','-','\0'}
l.03     int fd = open("fichero.txt",O_RDONLY);
l.04     int fd2 = dup(fd);
l.05     lseek(fd2, 3, SEEK_SET);
l.06     pread(fd,BUF+2,1,0);       - - 0 - \0
l.07     lseek(fd, 1, SEEK_CUR);
l.08     pread(fd,BUF,2,4);         4 5 0 - \0
l.09     printf("%s\n",BUF);
l.10     read(fd,BUF,1);
l.11     read(fd,BUF+3,2);
l.12     printf("%s",BUF); close(fd);
l.13   }
```

resposta xb.c => 4 5 0 - \0

2-bis: (0.5 puntos/points) Posible cambio: Se asume que "100" no está en la lista de peticiones.

En un determinado instante, la cola de peticiones de E/S para acceder a cilindros de un disco contenía las peticiones: [35, ~~100,~~ 360, 77, 125, 222]. Ya se han atendido las peticiones [77] (en primer lugar) y [35]. Indique cuál será el siguiente cilindro accedido si utilizamos los algoritmos:

Nun determinado instante, a cola de peticións de E/S para acceder a cilindros dun disco contiña as peticións: [35, ~~100,~~ 360, 77, 125, 222]. Xa se atenderon as peticións [77] (en primeiro lugar) e [35]. Indique cal será o seguinte cilindro accedido se utilizamos os algoritmos:

At a given time, the I/O requests queue for accessing cylinders within a disk contained the following requests [35, ~~100,~~ 360, 77, 125, 222]. Requests [77] (in first place) and [35] have already been attended. Which will be the next accessed cylinder according to the following algorithms: ?

SSTF: _____ ~~**100**~~ 125
C-SCAN: _____ **360**
SCAN: _____ ~~**100**~~ **125**

Asumiendo que el 100 no estuviese entre las peticiones de acceso a cilindros, los siguientes cilindros serían:
SSTF: Atendidos 77 y 35 ==> el más próximo sería 125

C-SCAN: El "ascensor" atiende peticiones al bajar, así que tras atender 77 y 35, se mueve hasta el primer cilindro 0, después hasta el cilindro más interno, para volver a "bajar" y atender peticiones en sentido "descendente", así que atenderá 360 en primer lugar.

SCAN: El "ascensor" atiende peticiones al bajar, y después volverá a subir, atendiendo también peticiones: así que tras atender 77 y 35, se mueve hasta el primer cilindro 0, desde donde comienza a atender peticiones en sentido "ascendente", así que atenderá 125 en primer lugar.

*Escriba **claramente** Apelidos y nombre en mayúsculas / Use capital letters for Surname and name*

Apellidos:_____**Nombre:**_____ **DNI:** _____

Sistemas Operativos - Grado Ingeniera Informática UDC. Xaneiro de 2021 Opción C

Parte E/S (1.5 puntos).

1: (0.5 puntos/points)

Respóndase V/F (Verdadero/Falso) o deje en blanco. Respuesta correcta: +0.1; Respuesta incorrecta: -0.1. Puntuación mínima en esta pregunta: 0.
Respóndase V/F (Verdadeiro/Falso) ou deixe en branco. Resposta correcta: +0.1; Resposta incorrecta: -0.1. Puntuación mínima para esta pregunta: 0.
Answer T/F (True/False) or leave as blank. Correct answer: +0.1; Wrong answer: -0.1. Minimum score for this question: 0.

A	B	C	D	E

← Responde V/F aquí : o/ou responde en blanco
either answer T/F here, or leave blank

A- **En una operación de salida de datos, el registro de datos de un controlador de dispositivo permite saber si el dato a enviar ya ha sido transferido al dispositivo**: __F___
Nunha operación de saída de datos, o rexistro de datos dun controlador de dispositivo permite saber se o dato a enviar xa foi transferido ao dispositivo.
During a data-output transference, the data register of a device controller permits us to know if the piece of data to be sent/written has already been transferred to the device.

B- **Considérese la estructura en capas del software de E/S. El device driver es el encargado de chequear si un proceso tiene permisos para abrir un fichero:** _F___
Considérese a estructura en capas do software de E/S. O device driver é o encargado de chequear se un proceso ten permisos para abrir un ficheiro.
Let us consider the layered structure of the I/O software. The device driver is in charge of checking if a process has enough permissions to open a file.

C- **Un disco duro con 2 caras y 16000 pistas por cara, tiene 16000 cilindros:** _V___
Un disco duro con 2 caras e 16000 pistas por cara, ten 16000 cilindros.
A hard disk with 2 sides and 16000 tracks in each side contains 16000 cylinders.

D- **Considerando la siguiente salida del ls: El *minor-number* del dispositivo /dev/sda es 0:** _V___
Considerando a seguinte saída do ls: O minor-number do dispositivo /dev/sda é 0.
Considering the following output of the *ls* command: The minor number of the device /dev/sda is 0.

```
user@beowulf:~$ ls -l /dev/sd*
 brw-rw---- 1 root disk  8,  0 nov 25 12:20 /dev/sda
 brw-rw---- 1 root disk  8,  1 nov 25 12:20 /dev/sda1
 brw-rw---- 1 root disk  8,  2 dic 31 01:25 /dev/sda2
 brw-rw---- 1 root disk  8, 16 nov 25 12:20 /dev/sdb
 brw-rw---- 1 root disk  8, 17 nov 25 12:20 /dev/sdb1
```

E- **Un controlador DMA configurado en modo ráfaga tiene menor prioridad de acceso al bus que otro configurado en modo robo de ciclos:** _F___
Un controlador DMA configurado en modo ráfaga ten menor prioridade de acceso ao bus que outro configurado en modo roubo de ciclos.
A DMA controller configured in burst mode has a lower priority when accessing the bus than other one configured in cycle-stealing mode.

2: (0.5 puntos/points)

En un determinado instante, la cola de peticiones de E/S para acceder a cilindros de un disco contenía las peticiones: [100, 360, 77, 155, 222]. Ya se han atendido las peticiones [100] (en primer lugar) y [77]. Indique cuál será el siguiente cilindro accedido si utilizamos los algoritmos:

Nun determinado instante, a cola de peticións de E/S para acceder a cilindros dun disco contiña as peticións: [100, 360, 77, 155, 222]. Xa se atenderon as peticións [100] (en primeiro lugar) e [77]. Indique cal será o seguinte cilindro accedido se utilizamos os algoritmos:

At a given time, the I/O requests queue for accessing cylinders within a disk contained the following requests [100, 360, 77, 155, 222]. Requests [100] (in first place) and [77] have already been attended. Which will be the next accessed cylinder according to the following algorithms: ?

SSTF: _____ 155 Tras 100 y 77, la más próxima será la 155

C-SCAN: _____ 360 Tras 100 y 77, --> el ascensor atiende peticiones al bajar... pero como no hay más hacia abajo, llegará al cilindro 0, de ahí se moverá al cilindro más interno y volverá a bajar atendiendo peticiones en sentido descendente --> la primera será la 360

SCAN: _____ 155 Tras 100 y 77 --> ascensor está bajando... llegará al cilindro y volverá a subir atendiendo peticiones --> la primera que se atenderá será la 155

3: (0.25 puntos/points)

Tenemos un disco que contiene 16384 (=16*1024) sectores. El disco tiene 1 plato y 2 caras. Cada sector contiene 128 bytes, y cada pista contiene 16 sectores. ¿Cuántos cilindros tiene?¿Cuántos bloques verá el S.O. si se formatea usando bloques de 2048 bytes?

*Temos un disco que contén 16384 (=16*1024) sectores. O disco ten 1 plato e 2 caras. Cada sector contén 128 bytes, e cada pista contén 16 sectores. ¿Cantos cilindros ten? ¿Cantos bloques verá o S.O. se se formatea usando bloques de 2048 bytes?*

We have a disk composed of 16384 (=16*1024) sectors. The disc has 1 platter and 2 sides. Each sector contains 128 bytes, and each track contains 16 sectors. How many cylinders are there on the disk? How many blocks will be seen by the O.S. if it is formated using blocks of 2048 bytes each?

Num cilindros/cylinders: 512

① Disco con 1 plato y 2 caras ⇒ cada cilindro tiene 2 pistas.
Disco con 16384 sectores × $\frac{1\ \text{pista}}{16\ \text{sectores}}$ = 1024 pistas × $\frac{1\ \text{cilindro}}{2\ \text{pistas}}$ = 512 cilindros.

Num bloques/blocks: 1024

② Bloques de 2048 bytes; cada sector tiene 128 bytes ⇒ cada bloque contiene 16 sectores.
16384 sectores × $\frac{1\ \text{bloque}}{16\ \text{sectores}}$ = 1024 bloques

4: (0.25 puntos/points)

El fichero "fichero.txt" contiene "012345678901234567890\n". ¿Cuál será el contenido de BUF[i] (i=1..4) TRAS ejecutar la instrucción en la línea 08 del siguiente programa?

El fichero "file.txt" contiene "012345678901234567890\n". Cuál será el contenido de BUF[i] (i=1..4) TRAS ejecutar la instrucción en la línea 08 del siguiente programa?

The file "file.txt" contains "012345678901234567890\n". Which will be the contents of BUF[i] (i=1..4) AFTER executing the intruction at line 08 in the following program?

After/tras lin.02: **BUF[i] =**	-	-	-	-	\0	
After/tras lin.08: **BUF[i] =**						<-- Fill your answer here <-- Contesta aquí
i =	0	1	2	3	4	

```
l.01 int main(){
l.02     char BUF[5]= {'-','-','-','-','\0'};
l.03     int fd = open("fichero.txt",O_RDONLY);
l.04     int fd2 = dup(fd);
l.05     lseek(fd2, 3, SEEK_SET);
l.06     pread(fd,BUF+3,1,0);          - - - 0 \0
l.07     lseek(fd, 1, SEEK_CUR);
l.08     pread(fd,BUF,2,1);            1 2 - 0 \0
l.09     printf("%s\n",BUF);
l.10     read(fd,BUF,1);
l.11     read(fd,BUF+1,2);
l.12     printf("%s",BUF); close(fd);
l.13   }
```

resposta xc.c => 1 2 - 0 \0