

Examen de Sistemas Operativos

Parte Práctica

Problema 1

De cierto sistema de archivos se conoce la tabla de nodos índice y el contenido de los primeros bloques de datos, mostrados en la imagen adjunta. Responda a las preguntas siguientes si un programa ejecuta el código mostrado, teniendo en cuenta que los bloques de datos se asignan de forma consecutiva, que la partición tenía, tras su formateo, 32 Gbytes disponibles y que cada dirección (tanto en nodos índice como en bloques) ocupa el número de bytes necesarios para direccionar todos los bloques de datos.

Tabla de Nodos Índice

Nodo índice	2	3	4	5	6
Tipo	DIR	DIR	REG	REG	REG
Enlaces	3	2	1	1	1
Tamaño	1024	1024	785	15	15000
P. Directos	1	2	3	4	5
	0	0	0	0	6
	0	0	0	0	7
P. Ind. Dob.	0	0	0	0	8

Bloques de Datos

Bloque	1	2
Contenido	.	2
	..	2
	directorio	3
	<u>fuentes.c</u>	4
	ejecutable	6
		archivo 5

```
1 fd=open ("ejecutable", O_RDONLY);
2 lseek (fd, 4000, SEEK_SET);
3 read (fd, &caracter, 1);
4 lseek (fd, 2000, SEEK_CUR);
5 read (fd, &caracter, 1);
```

1. Responda en la plantilla adjunta el máximo número de caracteres que se podrían escribir en un archivo, así como el número de archivos de ese tamaño que se podrían crear en la partición indicada, suponiendo para el cálculo que no existe ningún otro archivo. (0,8 puntos)

Answer: 32 Gbytes / 1024 bytes por bloque = 2^{25} bloques en la partición => 4 bytes

1024 bytes / 4 bytes = 256 direcciones por bloque

archivo tam. máx. = 3 bloques + $256 \cdot 256$ bloques = 65.539 bloques

máximo número de caracteres en un archivo = 65.539 bloques * 1.024 bytes/bloque = **67.111.936 bytes**

ocupación del archivo máximo => 65.539 bloques datos + (1 + 256) bloques direcciones = 65.796 bloques

número de archivos de tamaño máximo => 2^{25} bloques / 65796 bloques = 509,976 => **509 archivos**

2. Rellene las tablas de la plantilla adjunta indicando a qué estructuras de datos afectan la ejecución de las líneas de código indicadas. En el caso de afectar a nodos índice o bloques de datos, indique a qué nodos o bloques concretos se accede. (1,2 puntos)

Answer:

Tabla de Nodos Índice

Nodo índice	2	3	4	5	6
Tipo	DIR	DIR	REG	REG	REG
Enlaces	3	2	1	1	1
Tamaño	1024	1024	785	15	15000
P. Directos	1	2	3	4	5
	0	0	0	0	6
	0	0	0	0	7
P. Ind. Dob.	0	0	0	0	8

Bloques de Datos

Bloque	1	2	3	4	5-7
Contenido	.	.	Contenido de	Contenido de	Contenido de
	fuentes.c	archivo	ejecutable
	directorio	archivo			
	fuentes.c				
	ejecutable				
Bloque	8	9	10-21		
Contenido	9	10	Contenido de ejecutable		
		11			
		12			
		13			
		14			
		15			
		16			
		17			
		18			
		19			
		20			
		21			

	<i>TDA (SÍ o NO)</i>	<i>TAA (SÍ o NO)</i>	<i>Nodos Índice (indique valores)</i>	<i>Bloques de datos (indique valores)</i>
1- open ()	SÍ	SÍ	2,6	1
2- lseek ()	SÍ	SÍ	X	X
3- read ()	SÍ	SÍ	6	8,9,10
4- lseek ()	SÍ	SÍ	X	X
5- read ()	SÍ	SÍ	6	8,9,12

Problema 2

Teniendo en cuenta que el funcionamiento de la función `sprintf()` sólo se diferencia del de la función `printf()` en que la cadena de caracteres que genera es enviada a la variable pasada como primer argumento en vez de a la salida estándar, responda en la plantilla las cuestiones indicadas acerca del siguiente programa y su ejecución:

```
usuario@examenA$ cat -n fuente.c
 1  #include <stdio.h>
 2  #include <string.h>
 3  #include <fcntl.h>
 4  #include <unistd.h>
 5  #include <sys/types.h>
 6  #include <sys/wait.h>
 7
 8  int main () {
 9      int c=0, id, fd1, fd2;
10      char cadena[10];
11
12      close (1);
13      id=fork ();
14      fd1=open ("ARCHIVO", O_RDWR | O_CREAT | O_TRUNC, 0666);
15      c=c+write (1,"AAAA", 4);
16      wait (NULL);
17      if (id) {
18          fd2=open ("TOTAL", O_WRONLY | O_CREAT | O_TRUNC, 0666);
19          if (fork()) {
20              wait (NULL);
21              sprintf (cadena,"%d", c);
22              write (fd2, cadena, strlen (cadena));
23          } else {
24              wait (NULL);
25              c=c+write (1,"BBBB", 4);
26          }
27      } else
28          c=c+write (1,"CCCC", 4);
29
30      close (fd1); close (fd2);
31      return (0);
32  }
33
```

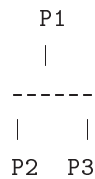
1. Indique los valores que tomarán las variables `fd1` y `fd2` durante la ejecución del programa. (0,3 puntos)

Answer:

`fd1=1`
`fd2=3`

2. Muestre los procesos que creará la ejecución del programa así como su relación jerárquica. (0,5 puntos)

Answer:



3. Muestre todas las tablas de descriptores de archivo y entradas de la tabla de archivos abiertos (indicando a qué archivo apuntan estas entradas) que se creen en algún momento durante la ejecución del programa con los valores previos a la ejecución de la línea 30, así como la vinculación entre todas ellas. (0,6 puntos)

Answer:

TDA Proceso1:

0=>Entrada estándar

1=>Entrada 1 de la TAA

2=>Salida estándar de errores

3=>Entrada 2 de la TAA

TDA Proceso2:

0=>Entrada estándar

1=>Entrada 3 de la TAA

2=>Salida estándar de errores

TDA Proceso3:

0=>Entrada estándar

1=>Entrada 1 de la TAA

2=>Salida estándar de errores

3=>Entrada 2 de la TAA

TAA Entradas:

1=>ARCHIVO

2=>TOTAL

3=>ARCHIVO

4. Muestre el contenido de los archivos `TOTAL` y `ARCHIVO` tras la finalización del programa ejecutable. (0,6 puntos)

Answer:

```
usuario@examenA$ more ARCHIVO TOTAL
::::::::::::::::::::
ARCHIVO => AAAABBBB
::::::::::::::::::::
::::::::::::
TOTAL => 4
::::::::::::
```