PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 2
(Primer semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 1 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **`INF239_0781_Ex2_P1_Buzón`** del **`Examen 2`** en PAIDEIA **antes de las 08:45**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_21.txt`. Por ejemplo, `20202912_21.txt`.

El algoritmo de *buddy system,* en una computadora binaria*,* no usa las operaciones aritméticas de multiplicación y división, sino las operaciones con bits que son mucho más rápidas. Consideremos un *buddy system* que maneja la asignación de memoria en un bloque de 512 KB (2^19 = 0x80000 bytes). Pero el bloque mínimo para solicitar la memoria es de 128 bytes. Es decir, no se puede solicitar 1 byte, ni 2 bytes, ni 4 bytes, etc., ni 64 bytes. Por eso, la primera entrada (número 0) de la tabla de *buddy system* contiene el puntero a la lista de bloques libres de tamaño 2^7 = 0x80 = 128 bytes. La siguiente entrada (número 1) de la tabla contiene el puntero a la lista de bloques libres de 2^8 = 0x100 = 256 bytes, etc., la última entrada (número 12) de la tabla contiene el puntero a la lista de bloques libres de 2^19 = 0x80000 = 512K bytes.

En el estado actual, solamente las entradas números 9, 10 y 11 contienen los punteros a las listas de bloques libres, el resto de entradas contienen el puntero nulo. La entrada número 9 apunta a la lista que tiene un solo bloque en la dirección 0x30000, la entrada número 10 apunta a la lista que también tiene un solo bloque en la dirección 0x00000 y la entrada número 11 apunta a la lista de un solo bloque en la dirección 0x40000.

a) (1 punto) Indique las direcciones hexadecimales de los tres bloques que serían los *buddies* de los bloques libres.

b) (2 puntos) Ejecute el programa proporcionado con 2 parámetros: su código estudiantil y el número 1:

```
$ ./2021-1_ex2 <su_código> 1
20NNNNNN: 0x...
```

En la salida del programa, para su código se proporciona un valor en hexadecimal que representa la solicitud de un bloque de memoria cuyo tamaño se indica en bytes. Indique, todo en hexadecimal, el tamaño del **bloque que será asignado** a esta solicitud y en **qué dirección**. No use las operaciones aritméticas de multiplicación, ni de división, sino las operaciones con los bits. Explíquelo.

c) (2 puntos) Después de esta asignación todos los bloques ocupados se liberan. Explique la fusión de todos los *buddies* en todas las listas indicando sus direcciones hexadecimales y modificaciones de las listas para obtener el único bloque de memoria libre en la lista correspondiente a la última entrada de la tabla.

Preparado por VK
con LibreOffice Writer en Linux Mint 20.1 “Ulyssa”

Profesores del curso: (0781) V. Khlebnikov

Lima, 14 de julio de 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 2
(Primer semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 2 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **`INF239_0781_Ex2_P2_Buzón`** del **`Examen 2`** en PAIDEIA **antes de las 09:30**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_22.txt`. Por ejemplo, `20202912_22.txt`.

Ejecute el programa proporcionado con 2 parámetros: su código estudiantil y "..":

```
$ ./2021-1_ex2 <su_código> ..
20NNNNNN: 013...
```

En la salida del programa, para su código se proporciona la cadena de referencia a las páginas de un sistema de la memoria virtual con 8 páginas y 4 marcos de página (de 0 a 3).

a) (1 punto) Presente el funcionamiento del algoritmo de reemplazo de páginas FIFO para la cadena dada e indique la cantidad de fallas de página sucedidas.

b) (2 puntos) Presente el funcionamiento del algoritmo de reemplazo de páginas óptimo para la cadena dada e indique la cantidad de fallas de página sucedidas.

c) (2 puntos) Presente el funcionamiento del algoritmo de reemplazo de páginas LRU para la cadena dada e indique la cantidad de fallas de página sucedidas.

Preparado por VK
con LibreOffice Writer en Linux Mint 20.1 "Ulyssa"

Profesores del curso: (0781) V. Khlebnikov

Lima, 14 de julio de 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 2
(Primer semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 3 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **`INF239_0781_Ex2_P3_Buzón`** del **`Examen 2`** en PAIDEIA **antes de las 10:15**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_23.txt`. Por ejemplo, `20202912_23.txt`.

MOS4E, Chapter 11, 11.8.2 Implementation of the NT File System, pp. 958,959:
"**Storage Allocation**

The model for keeping track of disk blocks is that they are assigned in runs of consecutive blocks, where possible, for efficiency reasons. For example, if the first logical block of a stream is placed in block 20 on the disk, then the system will try hard to place the second logical block in block 21, the third logical block in 22, and so on. One way to achieve these runs is to allocate disk storage several blocks at a time, when possible.

The blocks in a stream are described by a sequence of records, each one describing a sequence of logically contiguous blocks. For a stream with no holes in it, there will be only one such record. Streams that are written in order from beginning to end all belong in this category. For a stream with one hole in it (e.g., only blocks 0–49 and blocks 60–79 are defined), there will be two records. Such a stream could be produced by writing the first 50 blocks, then seeking forward to logical block 60 and writing another 20 blocks. When a hole is read back, all the missing bytes are zeros. Files with holes are called **sparse files**.

Each record begins with a header giving the offset of the first block within the stream. Next comes the offset of the first block not covered by the record. In the example above, the first record would have a header of (0, 50) and would provide the disk addresses for these 50 blocks. The second one would have a header of (60, 80) and would provide the disk addresses for these 20 blocks.

Each record header is followed by one or more pairs, each giving a disk address and run length. The disk address is the offset of the disk block from the start of its partition; the run length is the number of blocks in the run. As many pairs as needed can be in the run record. Use of this scheme for a three-run, nine-block stream is illustrated in Fig. 11-41.

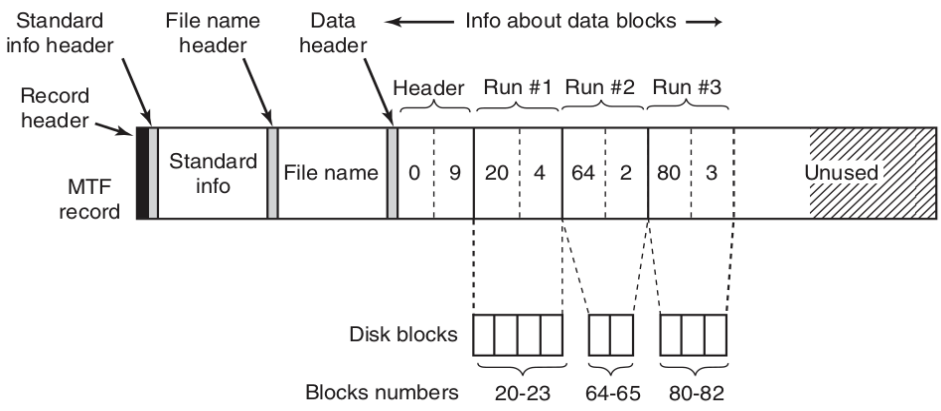

Figure 11-41. An MFT record for a three-run, nine-block stream.

In this figure we have an MFT record for a short stream of nine blocks (header 0–8). It consists of the three runs of consecutive blocks on the disk. The first run is blocks 20–23, the second is blocks 64–65, and the third is blocks 80–82. Each of these runs is recorded in the MFT record as a (disk address, block count) pair. How many runs there are depends on how well the disk block allocator did in finding runs of consecutive blocks when the stream was created. For an *n*-block stream, the number of runs can be anything from 1 through *n*."

a) (3 puntos) Según el texto presentado, si bloques de un archivo tienen la siguiente ubicación:

Offset	0	1	2	3	4	5	6	7	8	9	10
Bloque en disco	50	51	52	22	24	25	26	53	54	-	60

presente las entradas *run* (con el encabezado) grabadas en la MFT.

b) (2 puntos) Considere la Figura 11.41. Supongamos que el archivo creció y un 10mo bloque fue asignado al final del archivo. El número de este bloque es 66. ¿Cómo será ahora el registro de MFT?

Preparado por VK
con LibreOffice Writer en Linux Mint 20.1 "Ulyssa"

Profesores del curso: (0781) V. Khlebnikov

Lima, 14 de julio de 2021

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS
Examen 2
(Primer semestre de 2021)

Horario 0781: prof. V. Khlebnikov

Duración: 3 horas
Nota: **La presentación, la ortografía y la gramática influirán en la calificación.**
Puntaje total: 20 puntos

Pregunta 4 (5 puntos – 30 min.) El archivo de su respuesta debe estar en la carpeta **`INF239_0781_Ex2_P4_Buzón`** del **`Examen 2`** en PAIDEIA **antes de las 11:00**. Por cada 3 minutos de retardo son -2 puntos.
El nombre de su archivo debe ser <*su_código_de_8_dígitos*>`_24.txt`. Por ejemplo, `20202912_24.txt`.

Ejecute el programa proporcionado con 2 parámetros: su código estudiantil y "delta":

```
$ ./2021-1_ex2 <su_código> delta
20NNNNNN: ... bytes
```

En la salida del programa, para su código se proporciona el tamaño de un archivo en bytes.

Considere que fue creado el sistema de archivos ext2 con bloques de 1 KB. En este sistema de archivos, recien creado, se graba un archivo de tamaño indicado por el programa ejecutado. Indique qué bloques exactamente ocupará este archivo si el número del primer bloque libre es el número que se obtiene de los últimos 3 dígitos de su código incrementado en 100. ¿Y cómo el hecho de existencia de este archivo será indicado por el sistema de archivos?

Preparado por VK
con LibreOffice Writer en Linux Mint 20.1 "Ulyssa"

Profesores del curso: (0781) V. Khlebnikov

Lima, 14 de julio de 2021