

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**SISTEMAS OPERATIVOS**

**Examen 2**  
**(Primer semestre de 2023)**

Horarios 0781, 0782: prof. V. Khlebnikov

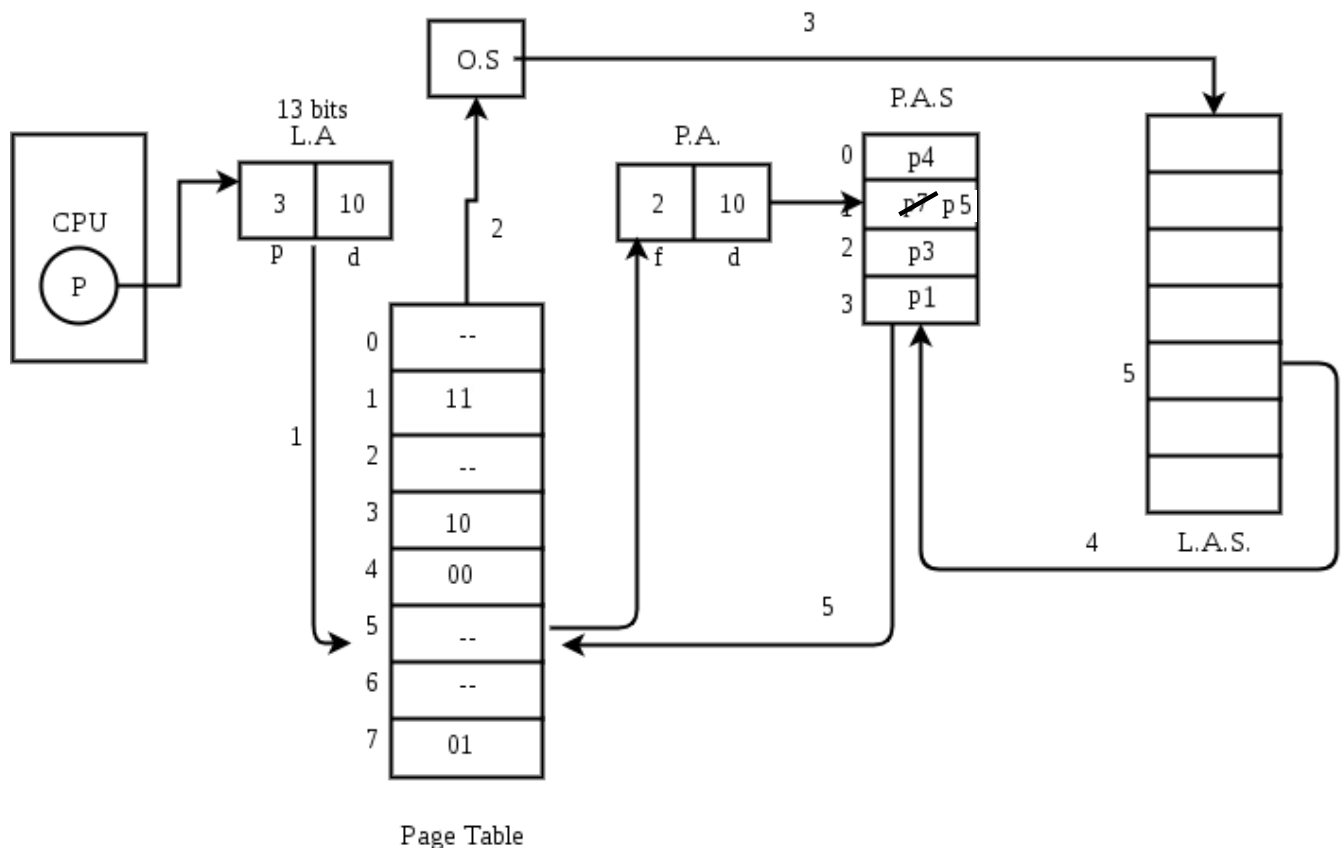
Duración: 3 horas

Notas: No está permitido el uso de ningún material o equipo electrónico.

**La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

**Pregunta 1 (6 puntos – 54 min.)** Considere la siguiente figura del proceso de la paginación por demanda:



**(a) (1 punto – 9 min.)** El proceso P, ejecutándose en el CPU, cambia su estado en los momentos marcados en la figura por las flechas 2 y 3, ¿al cuál?

**(b) (1 punto – 9 min.)** ¿Qué es L.A.S. en la figura?

**(c) (1 punto – 9 min.)** ¿Cuál será el estado final de la Page Table en la figura?

**(d) (1 punto – 9 min.)** En la figura están marcados con flechas solo los primeros 5 pasos del proceso de la paginación por demanda. En el paso 6 será enviada la señal al CPU que será procesada, y el proceso P cambiará su estado, ¿al cuál?

**(e) (2 puntos – 18 min.)** Considere los 8 dígitos decimales de su código del estudiante como los dígitos hexadecimales. Use los bits menos significativos de este número hexadecimal como los bits del campo *d* de L.A. de la figura. Presente L.A. completa (en binario y en hexadecimal) y P.A. completa (en binario y en hexadecimal) correspondientes a la figura presentada.

**Pregunta 2 (4 puntos – 36 min.)** Consider the virtual page reference string 1, 2, 3, 2, 4, 1, 3, 2, 4, 1 on a demand paged virtual memory system running on a computer system that main memory size of 3 pages frames which are initially empty. Let LRU, FIFO and OPTIMAL denote the number of page faults under the corresponding page replacements policy. Then

- (A) OPTIMAL < LRU < FIFO
- (B) OPTIMAL < FIFO < LRU
- (C) OPTIMAL = LRU
- (D) OPTIMAL = FIFO

¿Cuál es su respuesta y su justificación?

**Pregunta 3 (1 punto – 9 min.)** La tabla FAT y la estructura nodo-i lo tienen en común que ambos contienen los números de bloques de datos, o sea, ambos son unos índices, pero ¿cuál es la diferencia principal en las listas de estos índices?

**Pregunta 4 (1 punto – 9 min.)** The first sector (512 bytes) of a FAT filesystem is the *boot sector*. In Unix-like terminology this would be called the superblock. It contains some general information.

First an explicit example (of the boot sector of a DRDOS boot floppy).

```
00000000 eb 3f 90 49 42 4d 20 20 33 2e 33 00 02 01 01 00
00000020 02 e0 00 40 0b f0 09 00 12 00 02 00 00 00 00
00000040 00 00 00 00 00 00 00 00 00 70 00 ff ff 49 42
00000060 4d 42 49 4f 20 20 43 4f 4d 00 50 00 00 08 00 18
...
```

The 2-byte numbers are stored little endian (low order byte first).

Here the FAT12 version, that is also the common part of the FAT12, FAT16 and FAT32 boot sectors. See further below.

| Bytes   | Content   |
|---------|---|
| 0-2     | Jump to bootstrap (E.g. eb 3c 90; on i86: JMP 003E NOP.<br>One finds either eb xx 90, or e9 xx xx.<br>The position of the bootstrap varies.)  |
| 3-10    | OEM name/version (E.g. "IBM 3.3", "IBM 20.0", "MSDOS5.0", "MSWIN4.0".<br>Various format utilities leave their own name, like "CH-FOR18".<br>Sometimes just garbage. Microsoft recommends "MSWIN4.1".)<br>/* BIOS Parameter Block starts here */ |
| 11-12   | Number of bytes per sector (i?)   |
| 13      | Number of sectors per cluster (i?)  |
| 14-15   | Number of reserved sectors (i?)   |
| 16      | Number of FAT copies (i?)   |
| 17-18   | Number of root directory entries (i?)   |
| 19-20   | Total number of sectors in the filesystem (i?)  |
| 21      | Media descriptor type (i?)  |
| 22-23   | Number of sectors per FAT (i?)  |
| 24-25   | Number of sectors per track (i?)  |
| 26-27   | Number of heads (i?)  |
| 28-29   | Number of hidden sectors (i?)<br>Hidden sectors are sectors preceding the partition.<br>/* BIOS Parameter Block ends here */  |
| 30-509  | Bootstrap   |
| 510-511 | Signature 55 aa   |

En forma de una lista presente (en decimal) los 11 valores marcados con los signos “i?”

**Pregunta 5 (3 puntos – 27 min.)** Disponemos de un disco duro en el que queremos situar un sistema de archivos UNIX SYSTEM V, pero hemos introducido algunas modificaciones:

1. Los punteros a bloques de datos de los *i*-nodos han sido reducidos de forma que ahora cada *i*-nodo contiene únicamente 8 apuntadores directos a bloques de datos y un apuntador indirecto simple a bloques de datos.
2. El área de datos está formada por 256 bloques.
3. El tamaño del bloque es de 16 bytes.

**a) (1 punto – 9 min.)** Calcule cuál será el tamaño máximo de un archivo (en bytes) con las condiciones enumeradas.

**b) (2 puntos – 18 min.)** Disponemos de nuevo de un área de datos formada por 256 bloques, pero en este caso el tamaño del bloque es indeterminado. Se pide:

1. Justifique cuál será el número máximo de bloques que podrán ser utilizados para almacenar los datos del archivo.
2. Determine cuál será el tamaño mínimo de bloques necesario para que un único archivo pueda llegar a ocupar toda el área de datos.
3. Calcule cuál será el tamaño máximo de ese archivo en bytes.

**Pregunta 6 (5 puntos – 45 min.)** Se dispone de una versión simplificada del sistema de archivos UNIX SYSTEM V en el que la definición de *i*-nodo y de entrada de directorio han sido modificadas:

Estructura de un *i*-nodo:

| Bytes ocupados | Descripción                | Ejemplo        |
|----------------|----------------------------|----------------|
| 1              | Tipo de archivo y permisos | drwxr-x        |
| 1              | Id de dispositivo          | 0xXX           |
| 4              | Puntero directo            | 0x00000004     |
| 4              | Puntero directo            | 0x00000000     |
| 4              | Puntero indirecto simple   | 0x00000000     |
| 4              | Puntero indirecto doble    | 0x00000000     |
| 4              | Tamaño (bytes)             | 32000          |
| 10             | Fechas y otros             | 0XXXXXXXXXXXXX |

Estructura de una entrada de directorio:

| Bytes ocupados | Descripción       | Ejemplo    |
|----------------|-------------------|------------|
| 12             | Nombre de archivo | a.out      |
| 4              | <i>i</i> -nodo    | 0x00000006 |

A partir de esas estructuras y de los siguientes extractos de la tabla de *i*-nodos y del área de datos:

Tabla de *i*-nodos:

| <i>i</i> -nodo | byte    | 0         | 1          | 2          | 6          | 10         | 14         | 18    | 22    | 31 |
|----------------|---------|-----------|------------|------------|------------|------------|------------|-------|-------|----|
| 0              |         | reservado |            |            |            |            |            |       |       |    |
| 1              |         | reservado |            |            |            |            |            |       |       |    |
| 2              | drwxr-x | 10        | 0x00000006 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 1024  | ..... |    |
| 3              | -rwxr-x | 10        | 0x0000000F | 0x00000010 | 0x00000011 | 0x00000012 | 4053696    | ..... |       |    |
| 4              | -rwxr-x | 10        | 0x00000020 | 0x00000021 | 0x00000000 | 0x00000000 | 3072       | ..... |       |    |
| 5              | drwxr-x | 10        | 0x00000022 | 0x00000000 | 0x00000000 | 0x00000000 | 96         | ..... |       |    |
| 6              | drwxr-x | 10        | 0x00000023 | 0x00000000 | 0x00000000 | 0x00000000 | 96         | ..... |       |    |
| 7              | -rwxr-x | 10        | 0x00000010 | 0x00000012 | 0x00000006 | 0x0000000F | 4053696    | ..... |       |    |

Bloques del área de datos:

| bloque |            |            |            |            |            |            |        |            |  |
|--------|------------|------------|------------|------------|------------|------------|--------|------------|--|
| 6      | .          | 0x00000002 | ..         | 0x00000002 | vmunix     | 0x00000003 | tmp    | 0x00000005 |  |
|        | var        | 0x00000006 | etc        | 0x00000008 | lost+found | 0x00000001 | ...    | 0x0FF00000 |  |
| ...    |            |            |            |            |            |            |        |            |  |
| F      | 0x00001000 | 0x00001001 | 0x00001002 | 0x00001003 | 0x0000FFFF | 0x000010FF | ...    |            |  |
| 10     | 0x00002000 | 0x00002001 | 0x00002002 | 0x00002003 | 0x00002FFF | 0x000020FF | ...    |            |  |
| 11     | 0x000000A0 | 0x000000A1 | 0x000000A2 | 0x000000A3 | 0x000000B0 | 0x000000B1 | ...    |            |  |
| 12     | 0x00000014 | 0x00001002 | 0x00000013 | 0x00001003 | 0x00001004 | 0x00001005 | ...    |            |  |
| 13     | 0x00003000 | 0x00003001 | 0x0000CAFE | 0x00002903 | 0x000010FF | 0x000010FE | ...    |            |  |
| 14     | 0x00000700 | 0x00000701 | 0x00000702 | 0x00000903 | 0x0000F00F | 0x000007FE | ...    |            |  |
| ...    |            |            |            |            |            |            |        |            |  |
| 20     | 0x00002000 | 0x00002001 | 0x00002002 | 0x00002003 | 0x00002FFF | 0x000020FF | ...    |            |  |
| 21     | 0x00001000 | 0x00001001 | 0x00001002 | 0x00001003 | 0x00001004 | 0x00001005 | ...    |            |  |
| 22     | .          | ????????   | ..         | ????????   | a.out      | 0x00000004 | vmunix | 0x00000007 |  |

se pide:

- Si los bloques son de 2 KB, determine el tamaño de los siguientes componentes:
  - Tamaño máximo del sistema de archivo.
  - Tamaño máximo que puede ocupar un archivo.
  - Número máximo de archivos que se pueden almacenar en el sistema si la tabla de *i*-nodos ocupa los 5 bloques siguientes al superbloque.
  - Número de bloques del área de datos si tenemos un disco de 180 GB. Calcule también la densidad de *i*-nodo empleada (el tamaño promedio de un archivo) y comente el resultado.
- Bloque de disco en el que se encuentra el byte 3153940 del archivo /vmunix
- ¿Cuál es la ruta absoluta al archivo a.out?
- ¿Qué números de *i*-nodo deben aparecer en las entradas . y .. del bloque 22?
- ¿Qué inconsistencia existe entre las entradas del directorio raíz?



Preparado por VK  
con LibreOffice Writer en Linux Mint 21.1 "Vera"

Profesor del curso: V. Khlebnikov

Pando, 5 de julio de 2023