

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

**SISTEMAS OPERATIVOS**

**Examen 2**

**(Primer semestre de 2022)**

Horarios 0781, 0782

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\_078X\_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`.

Consideremos un ejemplo que muestra cómo funciona el algoritmo *buddy system*. En nuestro caso, entre los nodos del mismo tamaño, asignaremos el nodo con la menor dirección. Complete el texto y la tabla con los valores que están marcados con "...". En su respuesta pueden omitir la columna 2 de la tabla que indica los eventos.

Inicialmente tenemos un solo bloque en la dirección 0x0 y de tamaño MS que depende de su código de estudiante y se indica más abajo. Los valores son entre las siguientes posibilidades: 1MB = 0x100000 bytes, 2MB = 0x200000 bytes, 4MB = 0x400000 bytes, 8MB = 0x800000 bytes. Usted debe trabajar con su valor indicado.

Las solicitudes son en bytes y sus valores son diferentes para cada estudiante y se indican también más abajo. El bloque asignado por la solicitud denotaremos con la tupla (dirección, tamaño) (ambos en hexadecimal): (0x..., 0x...). Después de asignación y terminación de proceso se necesita indicar los bloques libres (dirección,tamaño): ... .

Se necesita presentar la siguiente tabla llena:

		Bloque asignado	Bloques libres
1.	Solicitud A	(0x...,0x...)	(0x...,0x...), ...
2.	Solicitud B	(0x...,0x...)	...
3.	Solicitud C	(0x...,0x...)	...
4.	Solicitud D	(0x...,0x...)	...
5.	Solicitud E	(0x...,0x...)	...
6.	C termina		...
7.	B termina		...
8.	Solicitud F	(0x...,0x...)	...
9.	Solicitud G	(0x...,0x...)	...
10.	D termina		...
11.	A termina		...
12.	G termina		...
13.	Solicitud H	(0x...,0x...)	...

14.	E termina		...
15.	Solicitud I	(0x...,0x...)	...
16.	Solicitud J	(0x...,0x...)	...
17.	Solicitud K	(0x...,0x...)	...
18.	Solicitud L	(0x...,0x...)	...
19.	H termina		...
20.	Solicitud M	(0x...,0x...)	...
21.	Solicitud N	(0x...,0x...)	...
22.	L termina		...
23.	F termina		...
24.	K termina		...
25.	Solicitud P	(0x...,0x...)	...
26.	I termina		...
27.	N termina		...
28.	J termina		...

20092344: MS=0x100000, A=0x21300, B=0x19800, C=0x15600, D=0x12800, E=0x2F600  
F=0x14900, G=0x1A600, H=0x65200, I=0xAE00, J=0xBF00, K=0x1B600  
L=0xA600, M=0x3E100, N=0x1D500, P=0x9E00

20102205: MS=0x800000, A=0x1FBB00, B=0xA9400, C=0xA5200, D=0xDAC00, E=0x1F5500  
F=0xE9C00, G=0xD6800, H=0x390300, I=0x49800, J=0x75800, K=0x89200  
L=0x59700, M=0x121800, N=0xF7E00, P=0x64F00

20130153: MS=0x200000, A=0x5B500, B=0x2C800, C=0x3DC00, D=0x24F00, E=0x51400  
F=0x3CD00, G=0x20E00, H=0xF1800, I=0x14500, J=0x1A500, K=0x38000  
L=0x11C00, M=0x58C00, N=0x38800, P=0x1F000

20150866: MS=0x200000, A=0x7E300, B=0x28C00, C=0x3F100, D=0x29700, E=0x60A00  
F=0x3C200, G=0x35C00, H=0xF5400, I=0x18E00, J=0x17600, K=0x2B100  
L=0x14E00, M=0x61200, N=0x34800, P=0x17D00

20151864: MS=0x100000, A=0x3C200, B=0x1F300, C=0x1E700, D=0x11200, E=0x2B600  
F=0x1C600, G=0x18300, H=0x5CB00, I=0xA300, J=0xCB00, K=0x1B900  
L=0x8C00, M=0x22E00, N=0x1C100, P=0xFB00

20155253: MS=0x100000, A=0x38400, B=0x1EB00, C=0x16C00, D=0x13800, E=0x39000  
F=0x10400, G=0x1E000, H=0x78400, I=0x8000, J=0xC000, K=0x1C700  
L=0x9400, M=0x36B00, N=0x1E800, P=0x8700

20155869: MS=0x400000, A=0xAC000, B=0x5C600, C=0x54800, D=0x56300, E=0xD8500  
F=0x65000, G=0x65100, H=0x1B2A00, I=0x28D00, J=0x3E800, K=0x52E00  
L=0x36400, M=0x8DE00, N=0x57300, P=0x28800

20160277: MS=0x200000, A=0x64300, B=0x3A000, C=0x39D00, D=0x3FA00, E=0x6B700  
F=0x31C00, G=0x36600, H=0xB5400, I=0x1C700, J=0x10C00, K=0x20200  
L=0x19E00, M=0x73E00, N=0x3AA00, P=0x1BE00

20160529: MS=0x800000, A=0x14BB00, B=0x87200, C=0xB0500, D=0x91200, E=0x14BF00  
F=0x9AF00, G=0xA3D00, H=0x290A00, I=0x67300, J=0x4F400, K=0xE4800  
L=0x57800, M=0x157200, N=0xDB000, P=0x51100

20161438: MS=0x100000, A=0x3F000, B=0x10B00, C=0x16500, D=0x12200, E=0x3B400  
F=0x1C900, G=0x1B100, H=0x62500, I=0xD700, J=0xD500, K=0x15500  
L=0xC900, M=0x3CA00, N=0x18500, P=0x9300

20165842: MS=0x100000, A=0x3D800, B=0x12B00, C=0x13A00, D=0x19D00, E=0x20600  
F=0x19A00, G=0x14200, H=0x59D00, I=0xD800, J=0xED00, K=0x12D00  
L=0x8E00, M=0x39E00, N=0x1F400, P=0x9200

20166399: MS=0x200000, A=0x52900, B=0x2F300, C=0x37C00, D=0x2CC00, E=0x53700  
F=0x2D100, G=0x30F00, H=0xD9700, I=0x1C900, J=0x15A00, K=0x37100  
L=0x1E700, M=0x72B00, N=0x30700, P=0x16000

20170064: MS=0x400000, A=0x90C00, B=0x41E00, C=0x41A00, D=0x46100, E=0xB7200  
F=0x6A400, G=0x60700, H=0x1A5200, I=0x36500, J=0x2E700, K=0x43500  
L=0x3CC00, M=0x88A00, N=0x43100, P=0x31300

20170089: MS=0x400000, A=0x9E800, B=0x43E00, C=0x4AC00, D=0x4BC00, E=0x9EC00  
F=0x59E00, G=0x5B300, H=0x1B1400, I=0x26C00, J=0x3ED00, K=0x78500  
L=0x23500, M=0xC9E00, N=0x79500, P=0x21100

20170607: MS=0x400000, A=0xBC000, B=0x63000, C=0x7B200, D=0x56600, E=0xE9500  
F=0x5F200, G=0x56800, H=0x1D2600, I=0x31700, J=0x20800, K=0x70E00  
L=0x23100, M=0x9D600, N=0x7DB00, P=0x2D500

20170923: MS=0x400000, A=0xC0E00, B=0x74600, C=0x63500, D=0x7C900, E=0x9A800  
F=0x41600, G=0x78D00, H=0x15B000, I=0x23100, J=0x27500, K=0x4BC00  
L=0x3E600, M=0xBF600, N=0x74400, P=0x38100

20170976: MS=0x200000, A=0x73D00, B=0x27000, C=0x2C900, D=0x2CE00, E=0x7F500  
F=0x2FA00, G=0x3E100, H=0x95F00, I=0x14900, J=0x11600, K=0x22C00  
L=0x19900, M=0x7E200, N=0x3B600, P=0x16200

20171189: MS=0x400000, A=0xCED00, B=0x4EF00, C=0x45400, D=0x45E00, E=0xACD00  
F=0x79100, G=0x68800, H=0x1B9900, I=0x30800, J=0x2FA00, K=0x64100  
L=0x21600, M=0x9DD00, N=0x7A900, P=0x23000

20171301: MS=0x800000, A=0x196400, B=0xBAC00, C=0x80500, D=0x9A300, E=0x151B00  
F=0xA5300, G=0x80C00, H=0x37E700, I=0x7A100, J=0x5AB00, K=0xCFC00  
L=0x6DF00, M=0x1E5300, N=0x95D00, P=0x53100

20171395: MS=0x200000, A=0x7EC00, B=0x20800, C=0x2C400, D=0x3EB00, E=0x58600  
F=0x2F900, G=0x2A200, H=0xB0800, I=0x18200, J=0x11600, K=0x32500  
L=0x11300, M=0x51600, N=0x36200, P=0x1E600

20171614: MS=0x800000, A=0x161100, B=0xCBFF00, C=0xFB200, D=0xCAE00, E=0x1C4400  
F=0x95100, G=0xD3700, H=0x2B3A00, I=0x7FA00, J=0x5C100, K=0xA8A00  
L=0x61E00, M=0x1D6900, N=0xA7900, P=0x5C800

20171878: MS=0x100000, A=0x2BD00, B=0x1EF00, C=0x17A00, D=0x1DB00, E=0x3DB00  
F=0x12500, G=0x10900, H=0x46200, I=0xCB00, J=0xFE00, K=0x1A400  
L=0x8100, M=0x34600, N=0x17800, P=0x8500

20172469: MS=0x400000, A=0xF1900, B=0x63200, C=0x7D800, D=0x6DF00, E=0xD1500  
F=0x57000, G=0x51A00, H=0x1A7900, I=0x22E00, J=0x32C00, K=0x48A00  
L=0x24C00, M=0x99500, N=0x72500, P=0x22900

20172545: MS=0x200000, A=0x73E00, B=0x2A200, C=0x3D900, D=0x24300, E=0x43200  
F=0x23200, G=0x3CE00, H=0xBA900, I=0x18900, J=0x10000, K=0x21A00  
L=0x1F600, M=0x57E00, N=0x3E800, P=0x16B00

20180214: MS=0x100000, A=0x22F00, B=0x17600, C=0x1C300, D=0x1E900, E=0x26400  
F=0x19600, G=0x1B800, H=0x56000, I=0x9400, J=0xEC00, K=0x12500  
L=0x9600, M=0x31600, N=0x1EC00, P=0xD100

20180230: MS=0x800000, A=0x1F9D00, B=0xF8900, C=0x89000, D=0xE4C00, E=0x105D00  
F=0x91800, G=0xA8400, H=0x233600, I=0x71600, J=0x5E500, K=0x8B000  
L=0x5C300, M=0x1B7700, N=0xA0800, P=0x75900

20180236: MS=0x100000, A=0x32200, B=0x1EA00, C=0x12000, D=0x11600, E=0x3FA00  
F=0x1A100, G=0x12400, H=0x56600, I=0xBC00, J=0xC500, K=0x14B00  
L=0xED00, M=0x31E00, N=0x1D100, P=0xC500

20180385: MS=0x100000, A=0x28B00, B=0x13C00, C=0x16D00, D=0x15700, E=0x2F700  
F=0x14F00, G=0x18200, H=0x73400, I=0xDC00, J=0xEC00, K=0x1D000  
L=0xB000, M=0x24400, N=0x14A00, P=0xD600

20180392: MS=0x100000, A=0x39C00, B=0x1A100, C=0x12700, D=0x1C000, E=0x29A00  
F=0x10500, G=0x1B000, H=0x57A00, I=0xF000, J=0xBC00, K=0x1D700  
L=0xFE00, M=0x36600, N=0x13900, P=0xCC00

20180448: MS=0x800000, A=0x174F00, B=0xF0800, C=0xED300, D=0xFF200, E=0x18A100  
F=0x8E500, G=0x91100, H=0x21CE00, I=0x4A600, J=0x46F00, K=0xB6600  
L=0x60800, M=0x199700, N=0xDBA00, P=0x48000

20180516: MS=0x400000, A=0xF7300, B=0x59E00, C=0x79F00, D=0x75100, E=0xDB200  
F=0x4C000, G=0x7C800, H=0x1E7B00, I=0x32200, J=0x32200, K=0x65A00  
L=0x33200, M=0xC0900, N=0x6B700, P=0x28B00

20180664: MS=0x100000, A=0x3C200, B=0x1F500, C=0x1DD00, D=0x18900, E=0x33600  
F=0x1B100, G=0x12E00, H=0x4C400, I=0x8E00, J=0xE300, K=0x1BF00  
L=0xF700, M=0x20500, N=0x1F100, P=0xF500

20181012: MS=0x100000, A=0x23700, B=0x12800, C=0x14400, D=0x1F200, E=0x21B00  
F=0x1F600, G=0x1C400, H=0x69D00, I=0x9400, J=0xB000, K=0x11A00  
L=0x8900, M=0x36C00, N=0x1E800, P=0xEC00

20181069: MS=0x800000, A=0x19B700, B=0x9BF00, C=0xDB600, D=0x80700, E=0x143B00  
F=0xAA000, G=0xF3600, H=0x388600, I=0x78000, J=0x7EF00, K=0xB5200  
L=0x74F00, M=0x140E00, N=0xA1F00, P=0x5D400

20181309: MS=0x800000, A=0x1FA600, B=0x99400, C=0xCED00, D=0xAFA00, E=0x13C300  
F=0xF3B00, G=0x98B00, H=0x251B00, I=0x68200, J=0x63400, K=0x96100  
L=0x63B00, M=0x1A9400, N=0xFCB00, P=0x78F00

20181923: MS=0x200000, A=0x4B800, B=0x3BB00, C=0x2E200, D=0x21900, E=0x7A200  
F=0x2A900, G=0x3F400, H=0xAB700, I=0x1F400, J=0x1DD00, K=0x34D00  
L=0x1C700, M=0x61500, N=0x34C00, P=0x11100

20182002: MS=0x200000, A=0x79C00, B=0x3AD00, C=0x33E00, D=0x35200, E=0x6C500  
F=0x34900, G=0x25500, H=0xFE200, I=0x1E300, J=0x1D900, K=0x34C00  
L=0x14100, M=0x42500, N=0x32700, P=0x16100

20182731: MS=0x800000, A=0x171800, B=0xB7F00, C=0xDC500, D=0xE8400, E=0x150300  
F=0xE8B00, G=0xC6A00, H=0x347E00, I=0x76700, J=0x5C500, K=0xB7D00  
L=0x56D00, M=0x1CCE00, N=0x93700, P=0x48800

20182778: MS=0x100000, A=0x22400, B=0x1F400, C=0x17B00, D=0x1A800, E=0x2E400  
F=0x1BA00, G=0x12500, H=0x4CD00, I=0x8200, J=0x9C00, K=0x17800  
L=0xE400, M=0x30700, N=0x1AF00, P=0x8000

20182841: MS=0x200000, A=0x48400, B=0x3E000, C=0x2A700, D=0x24D00, E=0x77E00  
F=0x37E00, G=0x37200, H=0x89700, I=0x17800, J=0x1E900, K=0x27A00  
L=0x12800, M=0x76400, N=0x2B900, P=0x17C00

20182970: MS=0x400000, A=0xEFB00, B=0x72600, C=0x45F00, D=0x67C00, E=0xEC600  
F=0x43E00, G=0x5A400, H=0x1D8F00, I=0x3F500, J=0x3A200, K=0x78C00  
L=0x30D00, M=0xEF500, N=0x7DE00, P=0x25400

20185462: MS=0x400000, A=0xE9E00, B=0x5D100, C=0x61200, D=0x5F600, E=0xC7000  
F=0x78E00, G=0x6EF00, H=0x1E7A00, I=0x20900, J=0x3AA00, K=0x68D00  
L=0x31400, M=0x92500, N=0x4B200, P=0x21800

20186008: MS=0x100000, A=0x2B600, B=0x17500, C=0x17F00, D=0x12000, E=0x3D800  
F=0x1DF00, G=0x1F700, H=0x61B00, I=0xC800, J=0xE600, K=0x1C100  
L=0xE900, M=0x37100, N=0x1CA00, P=0xE300

20186723: MS=0x400000, A=0xD1E00, B=0x7D300, C=0x6C500, D=0x76300, E=0xD4B00  
F=0x6B600, G=0x5FF00, H=0x1B6D00, I=0x3BA00, J=0x25300, K=0x54700  
L=0x29F00, M=0xBC300, N=0x6DC00, P=0x22B00

20191151: MS=0x400000, A=0x98800, B=0x59C00, C=0x40C00, D=0x5FB00, E=0xFD700  
F=0x53100, G=0x77A00, H=0x1C7A00, I=0x31500, J=0x3DF00, K=0x7FA00  
L=0x35800, M=0xB5800, N=0x64900, P=0x36D00

20191283: MS=0x200000, A=0x51700, B=0x3D600, C=0x26000, D=0x27800, E=0x7FD00  
F=0x26D00, G=0x3C900, H=0x9F300, I=0x18900, J=0x16D00, K=0x2F400  
L=0x1DF00, M=0x62300, N=0x30E00, P=0x1CE00

20191425: MS=0x100000, A=0x2E900, B=0x15F00, C=0x1A300, D=0x11000, E=0x3E000  
F=0x1CF00, G=0x10E00, H=0x47E00, I=0xFE00, J=0xFA00, K=0x19900  
L=0xC100, M=0x32900, N=0x13200, P=0xA200

20193232: MS=0x100000, A=0x35900, B=0x1E000, C=0x14C00, D=0x15B00, E=0x33A00  
F=0x17700, G=0x13100, H=0x44400, I=0x8700, J=0x9000, K=0x12000  
L=0xDE00, M=0x2AD00, N=0x10F00, P=0x9000



Preparado por VK  
con LibreOffice Writer en Linux Mint 20.3 “Una”

Profesor del curso: V. Khlebnikov

Lima, 6 de julio de 2022

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

**SISTEMAS OPERATIVOS**

**Examen 2**

**(Primer semestre de 2022)**

Horarios 0781, 0782

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\_078X\_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.

El espacio de direcciones virtuales es de 256 MB. En el sistema de paginación el tamaño de una página es 4 KB. Según su código de estudiante, se proporcionan 10 direcciones generadas durante la ejecución de un programa en una computadora que tiene la memoria de solo 16 KB:

20092344: 0x13295B8, 0x13295BC, 0x400078C, 0x13295C0, 0xF000C30  
 0x132B928, 0x4000790, 0x132B92C, 0x4001798, 0xF004C46

20102205: 0x132BC3C, 0x132BC40, 0x4000EBC, 0x132BC44, 0xF0001C8  
 0x132DFAC, 0x4000EC0, 0x132DFB0, 0x4001EC8, 0xF0041DE

20130153: 0x1332968, 0x133296C, 0x4000C28, 0x1332970, 0xF0008E0  
 0x1334CD8, 0x4000C2C, 0x1334CDC, 0x4001C34, 0xF0048F6

20150866: 0x1337A50, 0x1337A54, 0x4000884, 0x1337A58, 0xF000768  
 0x1339DC0, 0x4000888, 0x1339DC4, 0x4001890, 0xF00477E

20151864: 0x1337E38, 0x1337E3C, 0x40002E8, 0x1337E40, 0xF00065C  
 0x133A1A8, 0x40002EC, 0x133A1AC, 0x40012F4, 0xF004672

20155253: 0x1338B74, 0x1338B78, 0x4000508, 0x1338B7C, 0xF000B8C  
 0x133AEE4, 0x400050C, 0x133AEE8, 0x4001514, 0xF004BA2

20155869: 0x1338DDC, 0x1338DE0, 0x4000788, 0x1338DE4, 0xF000D8C  
 0x133B14C, 0x400078C, 0x133B150, 0x4001794, 0xF004DA2

20160277: 0x1339F14, 0x1339F18, 0x4000B38, 0x1339F1C, 0xF0001F8  
 0x133C284, 0x4000B3C, 0x133C288, 0x4001B44, 0xF00420E

20160529: 0x133A010, 0x133A014, 0x4000198, 0x133A018, 0xF000F88  
 0x133C380, 0x400019C, 0x133C384, 0x40011A4, 0xF004F9E

20161438: 0x133A39C, 0x133A3A0, 0x4000EAC, 0x133A3A4, 0xF000540  
 0x133C70C, 0x4000EB0, 0x133C710, 0x4001EB8, 0xF004556

20165842: 0x133B4D0, 0x133B4D4, 0x40004E8, 0x133B4D8, 0xF000FA0  
 0x133D840, 0x40004EC, 0x133D844, 0x40014F4, 0xF004FB6

20166399: 0x133B6FC, 0x133B700, 0x4000E8C, 0x133B704, 0xF000CB4  
 0x133DA6C, 0x4000E90, 0x133DA70, 0x4001E98, 0xF004CCA

20170064: 0x133C550, 0x133C554, 0x4000ED4, 0x133C558, 0xF00050C  
 0x133E8C0, 0x4000ED8, 0x133E8C4, 0x4001EE0, 0xF004522

20170089: 0x133C568, 0x133C56C, 0x40001FC, 0x133C570, 0xF000680  
 0x133E8D8, 0x4000200, 0x133E8DC, 0x4001208, 0xF004696

20170607: 0x133C76C, 0x133C770, 0x4000204, 0x133C774, 0xF000AE4  
0x133EADC, 0x4000208, 0x133EAE0, 0x4001210, 0xF004AFA

20170923: 0x133C8A8, 0x133C8AC, 0x4000118, 0x133C8B0, 0xF000470  
0x133EC18, 0x400011C, 0x133EC1C, 0x4001124, 0xF004486

20170976: 0x133C8E0, 0x133C8E4, 0x4000A44, 0x133C8E8, 0xF000774  
0x133EC50, 0x4000A48, 0x133EC54, 0x4001A50, 0xF00478A

20171189: 0x133C9B4, 0x133C9B8, 0x4000714, 0x133C9BC, 0xF00047C  
0x133ED24, 0x4000718, 0x133ED28, 0x4001720, 0xF004492

20171301: 0x133CA24, 0x133CA28, 0x40007F4, 0x133CA2C, 0xF0001CC  
0x133ED94, 0x40007F8, 0x133ED98, 0x4001800, 0xF0041E2

20171395: 0x133CA80, 0x133CA84, 0x40003EC, 0x133CA88, 0xF00079C  
0x133EDF0, 0x40003F0, 0x133EDF4, 0x40013F8, 0xF0047B2

20171614: 0x133CB5C, 0x133CB60, 0x4000190, 0x133CB64, 0xF000398  
0x133EECC, 0x4000194, 0x133EED0, 0x400119C, 0xF0043AE

20171878: 0x133CC64, 0x133CC68, 0x4000F38, 0x133CC6C, 0xF000E3C  
0x133EFD4, 0x4000F3C, 0x133EFD8, 0x4001F44, 0xF004E52

20172469: 0x133CEB4, 0x133CEB8, 0x400015C, 0x133CEBC, 0xF000224  
0x133F224, 0x4000160, 0x133F228, 0x4001168, 0xF00423A

20172545: 0x133CF00, 0x133CF04, 0x4000188, 0x133CF08, 0xF000DA0  
0x133F270, 0x400018C, 0x133F274, 0x4001194, 0xF004DB6

20180214: 0x133ECF4, 0x133ECF8, 0x40009E8, 0x133ECFC, 0xF000C0C  
0x1341064, 0x40009EC, 0x1341068, 0x40019F4, 0xF004C22

20180230: 0x133ED04, 0x133ED08, 0x400034C, 0x133ED0C, 0xF000DD8  
0x1341074, 0x4000350, 0x1341078, 0x4001358, 0xF004DEE

20180236: 0x133ED0C, 0x133ED10, 0x40005E0, 0x133ED14, 0xF000F10  
0x134107C, 0x40005E4, 0x1341080, 0x40015EC, 0xF004F26

20180385: 0x133EDA0, 0x133EDA4, 0x4000074, 0x133EDA8, 0xF0001C4  
0x1341110, 0x4000078, 0x1341114, 0x4001080, 0xF0041DA

20180392: 0x133EDA8, 0x133EDAC, 0x4000A04, 0x133EDB0, 0xF00016C  
0x1341118, 0x4000A08, 0x134111C, 0x4001A10, 0xF004182

20180448: 0x133EDE0, 0x133EDE4, 0x4000B84, 0x133EDE8, 0xF00079C  
0x1341150, 0x4000B88, 0x1341154, 0x4001B90, 0xF0047B2

20180516: 0x133EE24, 0x133EE28, 0x4000370, 0x133EE2C, 0xF000730  
0x1341194, 0x4000374, 0x1341198, 0x400137C, 0xF004746

20180664: 0x133EEB8, 0x133EEBC, 0x400015C, 0x133EEC0, 0xF000F04  
0x1341228, 0x4000160, 0x134122C, 0x4001168, 0xF004F1A

20181012: 0x133F014, 0x133F018, 0x40001E4, 0x133F01C, 0xF000C14  
0x1341384, 0x40001E8, 0x1341388, 0x40011F0, 0xF004C2A

20181069: 0x133F04C, 0x133F050, 0x4000EF4, 0x133F054, 0xF000D6C  
0x13413BC, 0x4000EF8, 0x13413C0, 0x4001F00, 0xF004D82

20181309: 0x133F13C, 0x133F140, 0x40000C0, 0x133F144, 0xF000A74  
0x13414AC, 0x40000C4, 0x13414B0, 0x40010CC, 0xF004A8A

20181923: 0x133F3A0, 0x133F3A4, 0x40008C0, 0x133F3A8, 0xF0003AC  
0x1341710, 0x40008C4, 0x1341714, 0x40018CC, 0xF0043C2

20182002: 0x133F3F0, 0x133F3F4, 0x40001C8, 0x133F3F8, 0xF000270  
0x1341760, 0x40001CC, 0x1341764, 0x40011D4, 0xF004286

20182731: 0x133F6C8, 0x133F6CC, 0x4000C6C, 0x133F6D0, 0xF000504  
0x1341A38, 0x4000C70, 0x1341A3C, 0x4001C78, 0xF00451A

20182778: 0x133F6F8, 0x133F6FC, 0x40002F8, 0x133F700, 0xF000354  
0x1341A68, 0x40002FC, 0x1341A6C, 0x4001304, 0xF00436A

20182841: 0x133F738, 0x133F73C, 0x40001E0, 0x133F740, 0xF000504  
0x1341AA8, 0x40001E4, 0x1341AAC, 0x40011EC, 0xF00451A

20182970: 0x133F7B8, 0x133F7BC, 0x400016C, 0x133F7C0, 0xF0006E0  
0x1341B28, 0x4000170, 0x1341B2C, 0x4001178, 0xF0046F6

20185462: 0x1340174, 0x1340178, 0x4000AE0, 0x134017C, 0xF000650  
0x13424E4, 0x4000AE4, 0x13424E8, 0x4001AEC, 0xF004666

20186008: 0x1340398, 0x134039C, 0x40003C8, 0x13403A0, 0xF000E60  
0x1342708, 0x40003CC, 0x134270C, 0x40013D4, 0xF004E76

20186723: 0x1340660, 0x1340664, 0x4000F28, 0x1340668, 0xF000960  
0x13429D0, 0x4000F2C, 0x13429D4, 0x4001F34, 0xF004976

20191151: 0x13417AC, 0x13417B0, 0x4000BCC, 0x13417B4, 0xF0003F4  
0x1343B1C, 0x4000BD0, 0x1343B20, 0x4001BD8, 0xF00440A

20191283: 0x1341830, 0x1341834, 0x4000DB0, 0x1341838, 0xF00094C  
0x1343BA0, 0x4000DB4, 0x1343BA4, 0x4001DBC, 0xF004962

20191425: 0x13418C0, 0x13418C4, 0x400094C, 0x13418C8, 0xF0000F4  
0x1343C30, 0x4000950, 0x1343C34, 0x4001958, 0xF00410A

20193232: 0x1341FD0, 0x1341FD4, 0x40000DC, 0x1341FD8, 0xF000DF4  
0x1344340, 0x40000E0, 0x1344344, 0x40010E8, 0xF004E0A

Presente el funcionamiento del algoritmo LRU e indique la cantidad de fallos de página que sucederán. Compare el resultado obtenido con el resultado del algoritmo óptimo.



Preparado por VK  
con LibreOffice Writer en Linux Mint 20.3 "Una"

Profesor del curso: V. Khlebnikov

Lima, 6 de julio de 2022



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

**SISTEMAS OPERATIVOS**

**Examen 2**

**(Primer semestre de 2022)**

Horarios 0781, 0782

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\_078X\_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.

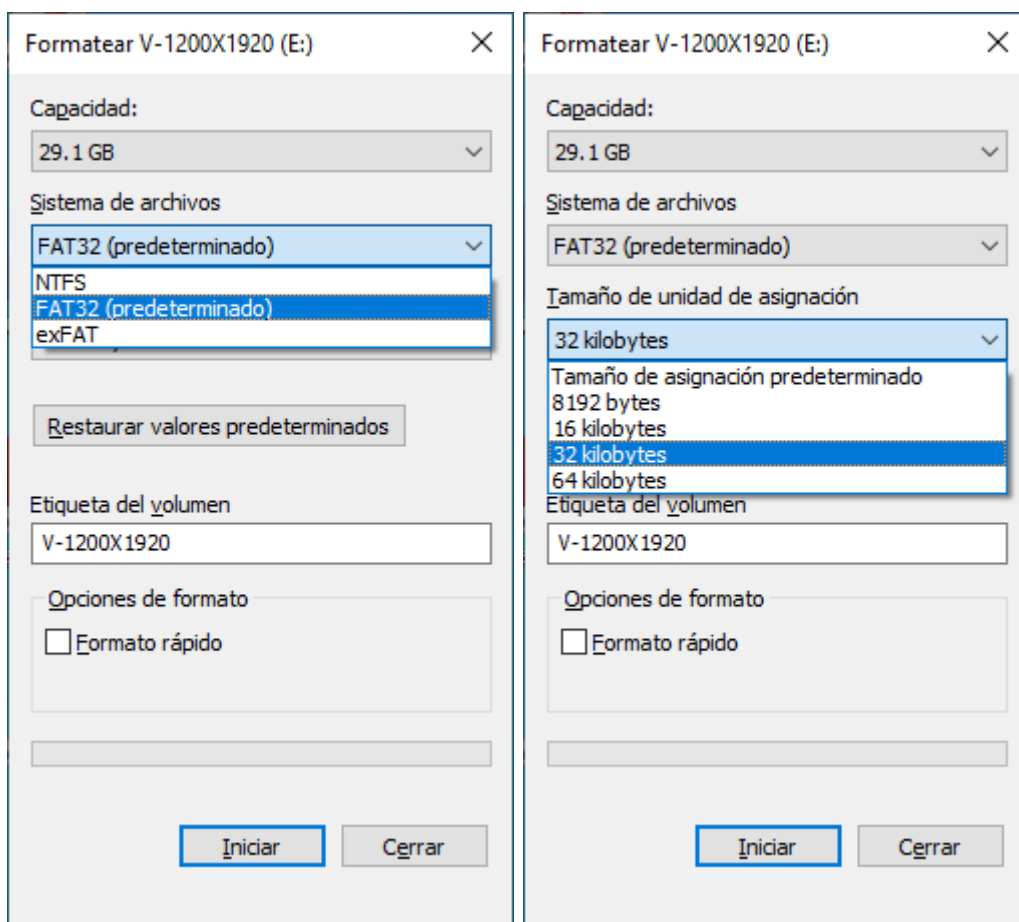
Una empresa, que tiene muchas oficinas de atención a clientes, decide instalar en sus oficina televisores grandes que constantemente proyectarán las imágenes desde una tarjeta de memoria. El proyecto realiza un equipo del departamento de tecnología de información (DTI) porque no se trata de un desarrollo de software. En el manual de los televisores se indica la recomendación de usar las tarjetas con el formato FAT32 y las imágenes en el formato JPG. El DTI adquiere los televisores, las tarjetas de 32 GB y prepara las imágenes en una computadora.

Cuando se prepara la tarjeta, el DTI selecciona la opción del sistema de archivo FAT32 porque es predeterminada y corresponde a la recomendación del manual del televisor.

Pero en la selección del tamaño de unidad de asignación (que usted conoce como *cluster*), el DTI entiende que este tamaño no debe ser predeterminado (el valor por defecto y genérico definido por la capacidad de la tarjeta) sino debe ser más conveniente para el despliegue rápido de las imágenes y dudan entre los tamaños de esta unidad en 32 y 64 KB.

Buscando en Internet y encontrando este

examen, el DTI solicita su consulta respecto a la mejor opción para el tamaño de unidad de asignación en el proceso de formateo de la tarjeta de memoria: ¿32 KB o 64 KB?



Usted analiza la tarjeta de memoria y tiene la siguiente información:

```
$ sudo fdisk /dev/sde
[sudo] contraseña:
```

```
Bienvenido a fdisk (util-linux 2.34).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.
```

```
Orden (m para obtener ayuda): p
Disco /dev/sde: 29,17 GiB, 31312576512 bytes, 61157376 sectores
Disk model: Storage Device
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x3c5330d7

Dispositivo Inicio Comienzo      Final Sectores Tamaño Id Tipo
/dev/sde1          32 61157375 61157344   29,2G  c W95 FAT32 (LBA)
```

O sea, en 61,157,344 sectores de 512 bytes está creado el sistema de archivo FAT32. Los primeros 32 sectores de esta tarjeta no se usan para este sistema de archivo.

Sabemos que cualquier sistema de archivos usa espacio para sus estructuras administrativas. Así los primeros 1456 sectores son reservados, después (en 61155888 sectores restantes) siguen la tabla FAT y su copia, y después comienza el área de *clusters* con el *cluster* 2 que contiene el directorio raíz. Todos los cálculos que se realizan son de la aritmética entera.

**a) (2 puntos)** ¿Cuántos sectores ocupará una tabla FAT en caso del tamaño de *cluster* en 32 KB y en 64 KB?

Para optimizar el uso del espacio en los directorios recomendaremos al DTI usar para todos los archivos solamente los nombres cortos en el formato 8.3 que sean los números consecutivos 00000000.JPG, 00000001.JPG, etc. También, para la rapidez del despliegue de las imágenes, recomendaremos limitar el tamaño de cualquier directorio (incluyendo el directorio raíz) a un solo *cluster*. Esto significa que en el directorio raíz habrán solo los nombres cortos de subdirectorios. Y las imágenes serán grabadas solo en subdirectorios.

**b) (1 punto)** ¿Cuántos archivos se puede grabar en un directorio en caso del tamaño de *cluster* en 32 KB y en 64 KB?

Suponiendo que el tamaño promedio de una imagen es de 512 KB + NNN KB, donde NNN es el número formado por los 3 últimos dígitos de su código de estudiante. Por ejemplo, si su código es 20159876, entonces el tamaño promedio de una imagen será 512 KB + 876 KB = 1388 KB = 1421312 bytes. Pero para la siguiente pregunta usted use su propio código.

**c) (2 puntos)** Si aplicamos nuestras recomendaciones, ¿cuántas imágenes podremos grabar en la tarjeta y cuántos subdirectorios del directorio raíz existirán en caso del tamaño de *cluster* en 32 KB y en 64 KB?



Profesor del curso: V. Khlebnikov

Preparado por VK  
con LibreOffice Writer en Linux Mint 20.3 "Una"

Lima, 6 de julio de 2022

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

**SISTEMAS OPERATIVOS**

**Examen 2**

**(Primer semestre de 2022)**

Horarios 0781, 0782

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\_078X\_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`.

Para la indicación de ubicación del contenido de archivo, EXT4, a diferencia de EXT2/EXT3, en vez del mecanismo de bloques indirectos, usa *extents*. (Un *extent* es un descriptor simple que representa un rango de bloques físicos contiguos.)

```
# echo Here is a new file >testfile
```

```
# ls -li testfile
```

```
918817 -rw-r--r-- 1 root root 19 2010-12-05 11:08 testfile
```

```
# istat /dev/mapper/elk-home 918817
```

```
inode: 918817
```

```
Allocated
```

```
Group: 112
```

```
Generation Id: 3173542730
```

```
uid / gid: 0 / 0
```

```
mode: rrw-r--r--
```

```
Flags:
```

```
size: 0
```

```
num of links: 1
```

```
[...]
```

Se obtiene la información sobre el sistema de archivo:

```
# fsstat /dev/mapper/elk-home
```

```
FILE SYSTEM INFORMATION
```

```
-----
```

```
File System Type: Ext3
```

```
[...]
```

```
CONTENT INFORMATION
```

```
-----
```

```
Block Range: 0 - 113971199
```

```
Block Size: 4096
```

```
Free Blocks: 13506529
```

```
BLOCK GROUP INFORMATION
```

```
-----
```

```
Number of Block Groups: 3479
```

```
Inodes per group: 8192
```

```
Blocks per group: 32768
```

```
[...]
```

```
Group: 112:
```

```
Inode Range: 917505 - ████████
```

```

Block Range: 3670016 - ████████
Layout:
Data bitmap: 3670016 - 3670016
Inode bitmap: 3670032 - 3670032
Inode Table: 3670048 - ████████
Data Blocks: 3670033 - 3670047, 3670560 - ████████
Free Inodes: 3281 (40%)
Free Blocks: 0 (0%)
Total Directories: 2
[...]

```

Uno de los cambios importantes en EXT4 consiste en que *inodes* ahora son de 256 bytes, mientras que en EXT2/EXT3 ellos fueron de 128 bytes. Esto significa que hay 16 *inodes* por bloque de 4KB en EXT4, por eso 8192 *inodes* por grupo de bloques deben ocupar ████████ bloques en el inicio de cada grupo.

**a) (1 punto)** Complete los 5 valores ocultos.

**b) (1 punto)** ¿En qué bytes y de qué bloque se encuentra el *inode* de nuestro archivo?

Ahora se puede desplegar el contenido de estos bytes:

```

00: A4 81 00 00 13 00 00 00 41 E3 FB 4C 41 E3 FB 4C
10: E3 FB 4C 00 00 00 00 00 00 00 01 00 08 00 00 00
20: 00 00 08 00 01 00 00 00 0A F3 01 00 04 00 00 00
30: 00 00 00 00 00 00 00 00 01 00 00 00 3F 88 3A 00
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60: 00 00 00 00 4A 6B 28 BD 00 00 00 00 00 00 00 00
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80: 1C 00 00 00 F0 E4 D4 D2 F0 E4 D4 D2 F0 E4 D4 D2
90: 41 E3 FB 4C F0 E4 D4 D2 00 00 00 00 00 00 00 00
A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

La estructura de *inode* es la siguiente: los primeros 2 bytes (0x81A4, en little-endian) es *i-mode* que indica el archivo regular con los permisos `rw-r--r--`; los bytes 2-3 es *uid*; los bytes 4-7 es el tamaño del archivo en bytes; los bytes 8-0xB, 0xC-0xF, 0x10-0x13, y 0x14-0x17 son los últimos tiempos de acceso, de cambio de *inode*, de modificación y de eliminación, respectivamente; los bytes 0x18-0x19 es *gid*; los bytes 0x1A-0x1B es *hard link count*; los bytes 0x1C-0x1F es *block count*; los bytes 0x20-0x23 es *inode flags* cuyo valor 0x80000 indica que *inode* usa *extents*, lo que significa que en los 60 bytes a partir del offset 0x28 no están los 15 punteros a los bloques directos e indirectos sino el árbol de *extents*.

Las estructuras de *extents* son de 12 bytes, y se podría esperar a tener 5 *extents* en cada nodo. Pero los primeros 12 bytes del área de *extents* están ocupados por un *extent header*, por eso en el *inode* se puede guardar solamente 4 *extents*. En el *extent header* se guardan los siguientes valores: el número mágico (2 bytes), la cantidad de *extents* (2 bytes), la cantidad máxima de *extents* (2 bytes), la profundidad del árbol (2 bytes), el ID de generación (4 bytes).

**c) (1 punto)** Indique los valores hexadecimales guardados en el *extent header* del nuestro archivo.

La estructura del mismo *extent* es la siguiente: el número del primer bloque del archivo (del bloque lógico) que forma este *extent* (4 bytes), la cantidad de bloques en el *extent* (2 bytes), los 16 bits más significativos del número del bloque físico, los 32 bits menos significativos del número del bloque físico.

**d) (1 punto)** ¿En qué bloque del disco están guardados los datos del nuestro archivo?

Después de eliminar el nuestro archivo el contenido del mismo inode se modificó:

```
00: A4 81 00 00 00 00 00 00 35 F6 FB 4C 35 F6 FB 4C
10: 35 F6 FB 4C 00 00 00 00 00 00 01 00 08 00 00 00
20: 00 00 08 00 01 00 00 00 0A F3 00 00 04 00 00 00
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60: 00 00 00 00 CC 6B 28 BD 00 00 00 00 00 00 00 00
70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
80: 1C 00 00 00 E8 A0 00 04 E8 A0 00 04 E8 A0 00 04
90: 35 F6 FB 4C E8 A0 00 04 00 00 00 00 00 00 00 00
A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

**e) (1 punto)** ¿Qué posibilidades hay para recuperar nuestro archivo eliminado según la información en este inode? Explique.



Preparado por VK  
con LibreOffice Writer en Linux Mint 20.3 “Una”

Profesor del curso: V. Khlebnikov

Lima, 6 de julio de 2022