

340 A3

UPI: enau831

ID: 768485633

Name: Etienne Naude

Question 1

16 blocks

4 KiB blocks size

4 Byte block number

1a

65,536 bytes

$16 * 4096 =$

65,536 bytes

1b

4,259,840 bytes

$65,536 + 1024 * 4096 =$

4,259,840

1c

4,299,227,136 bytes

$65536 +$

$1024 * 4096 +$

$(1024)^2 * 4096 =$

4,299,227,136

1d

4,402,345,738,240 bytes

$65536 +$

$1024 * 4096 +$

$(1024^2) * 4096 +$

$(1024^3) * 4096 =$

4,402,345,738,240 bytes

1e

5

aim : 1152921504606846976

max3: 4402345738240

max4: 4508001973108736

max5: 4616194020400496640

aim is > 4 but aim < 5 so the number of levels needed is 5

1f

2

direct twice

1g

5

single indirect + double indirect

1h

6

double indirect twice

1i

10

overwrites everything

5 for reading the data

then the blocks have been loaded into memory

then change the blocks in memory, change the inode

and write blocks to the disk

so another 5

1j

14

Assuming this appends data as the specs doesn't say "This overwrites data."

2 to first block then check the file size meta data

6 read

6 write

1k

These assumptions change how many accesses will be needed, for example if the inode were not loaded into memory they it would have to be found and read before anything else or if the file access times needed to be written they would increase the number of accesses.

Question 2

Intel core i5-10210u

Nearly all 64bit processors are backwards compatible with 32bit addresses. This is because many programs don't need to use the whole 32-bit address space, let alone the 64-bit one.

They need to be reprogrammed to take advantage of the full 64-bit address space this is especially relevant for operating systems, which needs to allocate virtual to real addresses.

Question 3

16GB

8KB

1MB

$16\text{GB}/8\text{KB} = 2,000,000$ total frames

bit map

$2,000,000/8 = 250,000$ bytes (one bit per frame)

linked list

$16\text{GB}/1\text{MB} = 16,000$ chunks

$16,000 * 32 = 512,000$ bits = 64,000 bytes

The bitmap would normally be in kernel memory so for that approach all of it. But for the linked list, only the start would be in kernel memory

the extents version would use less space than the bitmap but slightly more than the linked list method

Question 4

549,755,813,888 bytes

$4398.046511104 \text{ Gbits} = 549,755,813,888 \text{ bytes}$

$2^{32} * 1024$

not realistic at all. Most programmes wont need it.

Question 5

649.51ns

$(5000151 * 0.0001) + (151 * .99)$

649.5051ns

Question 6

3 Levels

8,388,609 total pages.

There are 3 levels, first level has 1 page, next level has 65536 pages and the last level has the rest needed to point to all the data.

64 bit addresses

2^{22} byte pages

2^6 byte of entry

$2^{(22-6)} = 65536$ entries per page table = 16 bits

$64-22+3 = 39$ bits to encode a page

$39//16 = 3$

total levels is 3

$1+65,536+$

65,536

$2^{(39-16)} = 8388608$

1

--> 65536 --> data

--> --> --> 8323072 --> data

8,388,609 total pages.

Question 7

40 seconds

Time per fault * faults == $(4.0e+10)ns$ == (40s)

$$2.0e+4 * 2.0e+6$$

$$= 4.0e+10$$

There for there are 20s of non-errors.

$$60s - 40s$$

$$= 20s$$

So there are $(2.0e+10)$ non error instructions

Double the memory, half the errors.

$$1.0e+4 \setminus * 2.0e+6$$

$$= (2.0e+10)ns \text{ (20s)}$$

$$20s \text{ (faults)} + 20s \text{ (normal running)} + 10,000ns =$$

$$40,000,000,000+10,000ns = 40,000,010,000$$

≈ 40 seconds

This is a significant reduction

Question 8

8a - FIFO

12 faults

	1	2	3	4	5	4	3	2	1	6	7	1	2	3	7	5	1
0	1	=	=	=	5	=	=	=	=	=	=	=	2	=	=	=	=
0	=	2	=	=	=	=	=	=	1	=	=	=	=	3	=	=	=
0	=	=	3	=	=	=	=	=	=	6	=	=	=	=	=	5	=
0	=	=	=	4	=	=	=	=	=	=	7	=	=	=	=	=	1

8b - LRU

11 faults

	1	2	3	4	5	4	3	2	1	6	7	1	2	3	7	5	1
0	1	=	=	=	5	=	=	=	1	=	=	=	=	=	=	5	=
0	=	2	=	=	=	=	=	=	=	=	=	=	=	=	=	=	1
0	=	=	3	=	=	=	=	=	=	=	7	=	=	=	=	=	=
0	=	=	=	4	=	=	=	=	=	6	=	=	=	3	=	=	=

8c - LFU

11 faults

	1	2	3	4	5	4	3	2	1	6	7	1	2	3	7	5	1
0	1	=	=	=	5	=	=	=	1	=	=	=	=	=	=	=	=
0	=	2	=	=	=	=	=	=	=	6	7	=	2	=	=	=	=
0	=	=	3	=	=	=	=	=	=	=	=	=	=	=	=	=	=
0	=	=	=	4	=	=	=	=	=	=	=	=	=	=	7	5	=

8d - Optimal

10 faults

	1	2	3	4	5	4	3	2	1	6	7	1	2	3	7	5	1
0	1	=	=	=	5	=	=	=	=	=	=	=	=	=	=	=	=

	1	2	3	4	5	4	3	2	1	6	7	1	2	3	7	5	1
0	=	2	=	=	=	=	=	=	=	6	7	=	=	=	=	=	=
0	=	=	3	=	=	=	=	=	=	=	=	=	2	3	=	=	=
0	=	=	=	4	=	=	=	=	1	=	=	=	=	=	=	=	=