LOUISIANA STATE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

CSC 4103: Operating Systems
Prof. Aisha Ali-Gombe
Lilly Moreau

Written Assignment Rules

(1) Consider a system with virtual address spaces for processes of 128 pages of 1,024 bytes each. The system has a physical memory of 64 frames.

(a) How many bits are there in a virtual address? How many bits make up the page number and how many make up the offset?

i Total virtual address space = # page * page size = $128 * 1{,}024 = 131{,}072\ bytes$

1 $Bytes = 2^{bits} = 131{,}072$

2 So virtual address is 17 bits long

ii # Page: $log_2(\#\ page) = log_2(128) = 7\ bits$

iii Offset: Virtual address bits – page bits = 17 - 7 = 10 bits

(b) How many bits are there in a physical address? How many bits make up the page number and how many make up the offset?

i Total physical address space = # frames * page size = $64 * 1{,}024 = 65{,}536\ bytes$

1 $Bytes = 2^{bits} = 65{,}536$

2 $2^{16} = 65{,}536$ so 16 bits for physical address

ii Page number: $log_2(\#\ frames) = log_2(64) = 6\ bits$

iii Offset: physical address – page # = 16 - 6 = 10 bits

Assume that a system uses demand paging and has *m* frames of memory available to processes. Now assume that a page reference string for a process has length *p*, and *n* distinct page numbers occur in it. All of the *m* frames of available memory are empty.

(c) What is a lower bound on the number of page faults for **any** page replacement algorithm?

i The lower bound on the number of page faults for any page replacement algorithm is equal to the number of distinct pages so *n*. This is because every page will create a page fault the first time it is referenced which will total *n* times.

(d) What is an upper bound on the number of page faults for **any** page replacement algorithm?

i The upper bound on the number of pages faults is going to equal the process length so *p*. It will be of length p because every time a page needs to be added to memory it would cause a page fault in the worse case scenario.

ii If $m >= n$, it would at most be equal to n because once added to memory it would not have to be replaced. This is unlikely but still a possibility.

(3) Consider the following page reference string:

7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1.

Assuming demand paging with three frames of memory, how many page faults would occur for the following replacement algorithms? Show your work!

(a) LRU replacement

Looking for:	*7*	*2*	*3*	*1*	*2*	*5*	*3*	*4*	*6*	*7*	*7*	*1*	*0*	*5*	*4*	*6*	*2*	*3*	*0*	*1*
F1	**7**	7	7	**1**	1	1	**3**	3	3	**7**	**7**	7	7	**5**	5	5	**2**	2	2	**1**
F2	-	**2**	2	2	**2**	2	2	**4**	4	4	4	**1**	1	1	**4**	4	4	**3**	3	3
F3	-	-	**3**	3	3	**5**	5	5	**6**	6	6	6	**0**	0	0	**6**	6	6	**0**	0
Page Fault?	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y

18 Page Faults

(b) FIFO replacement

Looking for:	*7*	*2*	*3*	*1*	*2*	*5*	*3*	*4*	*6*	*7*	*7*	*1*	*0*	*5*	*4*	*6*	*2*	*3*	*0*	*1*
F1	**7**	7	7	**1**	1	1	1	1	**6**	6	6	6	**0**	0	0	**6**	6	6	0	0
F2	-	**2**	2	2	**2**	**5**	5	5	5	**7**	**7**	7	**7**	**5**	5	5	**2**	2	2	**1**
F3	-	-	**3**	3	3	3	**3**	**4**	4	4	4	**1**	1	1	**4**	4	4	**3**	3	3
Page Fault?	Y	Y	Y	Y	N	Y	N	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y

17 Page Faults

(c) Optimal replacement

Looking for:	7	2	3	1	2	5	3	4	6	7	7	1	0	5	4	6	2	3	0	1
F1	**7**	7	7	**1**	1	1	1	1	1	1	1	**1**	1	1	1	1	1	1	1	**1**
F2	-	**2**	2	2	**2**	**5**	5	5	5	5	5	5	5	**5**	**4**	**6**	**2**	**3**	3	3
F3	-	-	**3**	3	3	3	**3**	**4**	**6**	**7**	**7**	7	**0**	0	0	0	0	0	**0**	0
Page Fault?	Y	Y	Y	Y	N	Y	N	Y	Y	Y	N	N	Y	N	Y	Y	Y	Y	N	N

13 Page Faults

(4) Assume a filesystem uses inodes with 12 direct block numbers, 1 indirect, 1 double indirect, 1 triple indirect. Assume that blocks hold 4096 bytes and block numbers consume 8 bytes.

(a) What is the maximum size file (in blocks) that can be accommodated?

i $4096/8 = 512$

ii Direct blocks + single indirect + double indirect + triple indirect = total

iii $12 + 1 * 512 + 1 * 512^2 + 1 * 512^3 = 134480396$ blocks

(b) How many bits are required for offsets into a file of the maximum size?

i # Offset bits = $\log_2(4096) = 12$