Max Harris
CSCI 3753

Problem Set 4

1.
a. Use best fit, worst fit, first fit & next fit. Some of these may actually cause more external fragmentation over time when talking about specifics, however they are general approaches for solving fragmentation.

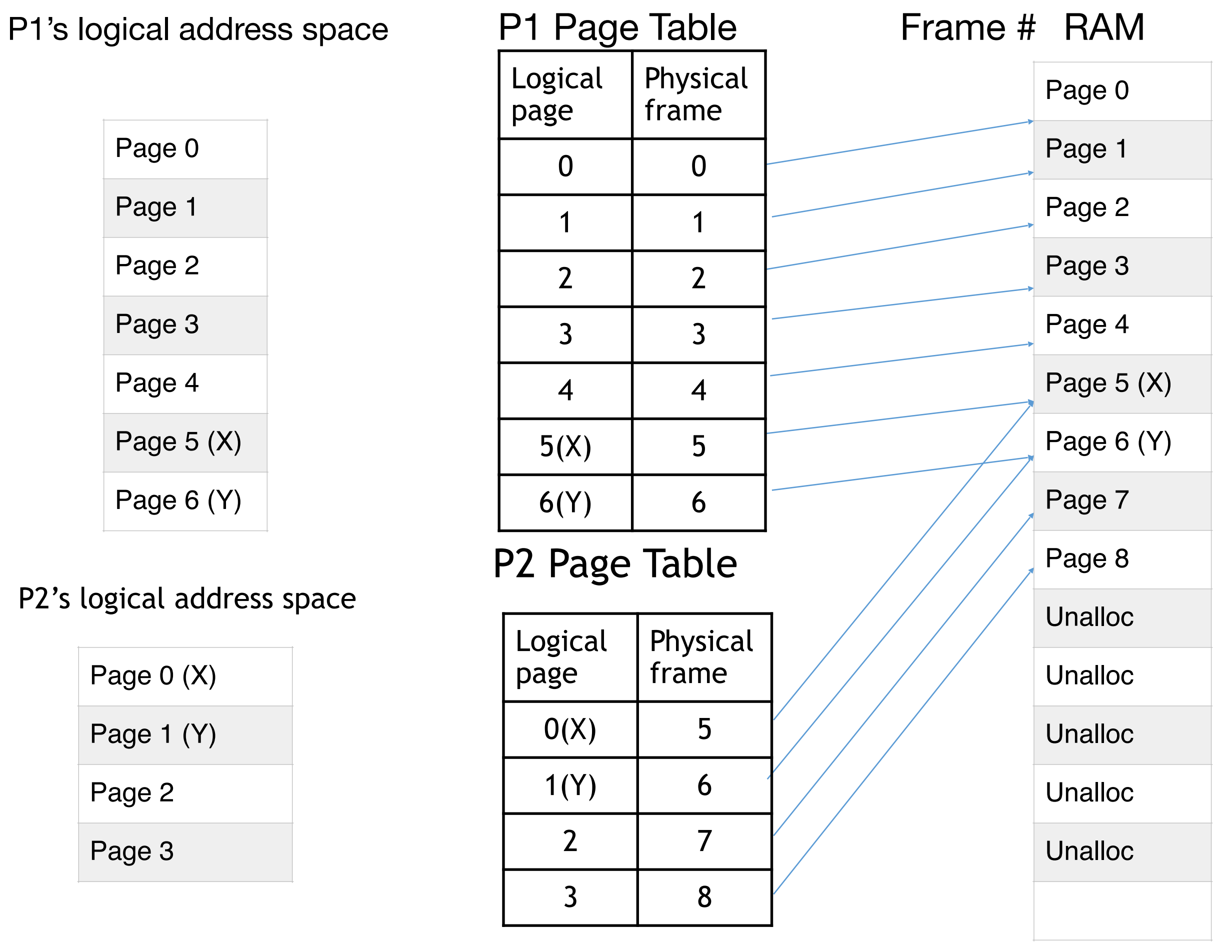
b. Use paging to divide the logical address space into fixed size pages. This allows each process to be split into pages that are distributed non-contagiously throughout memory. So when a new process wants to be allocated we find and set any number of pages for that process. This does solve external fragmentation, but then we may get some internal fragmentation if our process size is bigger than 1 page size.

c. Use variable-sized segmentation. This is similar to paging, but now we can divide a process based on some logical criteria, ie code, data. Furthermore, a process can subdivide its code into functional segments. Even though this solves external fragmentation, segmentation is very complex and difficult to manage.

2.

a.	2200
b.	1000
c.	2200
d.	2200
e.	2200
f.	2200

3. The two pages of shared code 'X' and 'Y' are pages 5&6 from P1 and pages 0&1 from P2 are common to both address spaces.



P1's logical address space

Page 0
Page 1
Page 2
Page 3
Page 4
Page 5 (X)
Page 6 (Y)

P1 Page Table

Logical page	Physical frame
0	0
1	1
2	2
3	3
4	4
5(X)	5
6(Y)	6

P2's logical address space

Page 0 (X)
Page 1 (Y)
Page 2
Page 3

P2 Page Table

Logical page	Physical frame
0(X)	5
1(Y)	6
2	7
3	8

Frame # RAM

Page 0
Page 1
Page 2
Page 3
Page 4
Page 5 (X)
Page 6 (Y)
Page 7
Page 8
Unalloc
Unalloc
Unalloc
Unalloc
Unalloc

4.

T	= 1ns	= TLB Hit
M	= 10ns	= Memory Read
D	= 10,000,000ns (10ms)	= Disk Read
p_TLB	= 0.9	= probability TLB hit
p	= 0.0001	= probability Page Fault (TLB miss)

(D • p) + (T • p_TLB) + ((1 - p_TLB - p) • M)
Average = (10,000,000 • 0.0001) + (1 • .9) + ((1 - 0.9 - 0.0001) • 10) = 1,001.899 ns

5.

LRU will replace the page whose most recent request was earliest. The LRU algorithm avoids the situation of replacing a frequently used variable if the approaching variable appears to be closer to the current reading variable. This means that as long as a process shows some locality then it will see better performance from increased memory.
Example: (using data from #6 but with a queue size of 4)
3 2 4 3 4 2 2 3 4 5 6 7 7 6 5 4 5 6 7
2 1

3	3	3							3	3	7								7	7
	2	2							2	6	6								6	6
		4							4	4	4								4	2
									5	5	5								5	1

LRU queue size 4: 8 page faults

6.a FIFO = 12 faults.

3	3	3							5	5	5				4	4	4	7	7	7
	2	2							2	6	6				6	5	5	5	2	2
		4							4	4	7				7	7	6	6	6	1

3 2 4 3 4 2 2 3 4 5 6 7 7 6 5 4 5 6 7 2
1

b. OPT = 10 faults.

3	3	3							5	5	5				5			7	7	7
	2	2							2	6	6				6			6	2	2
		4							4	4	7				4			4	4	1

3 2 4 3 4 2 2 3 4 5 6 7 7 6 5 4 5 6 7
2 1

c. LRU = 10 faults.

3	3	3							3	6	6				6			6	6	1
	2	2							5	5	5				5			5	2	2
		4							4	4	7				4			7	7	7

3 2 4 3 4 2 2 3 4 5 6 7 7 6 5 4 5 6 7
2 1

OPT and LRU both tie for 10 page faults

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

3	3	3							3	3	3									1
	2	2							2	2	2									2
		4							4	4	4									4
									5	5	5									5
										6	6									6
											7									7

We get 7 page faults using a dynamic paging working-set algorithm.