Lucas Dachman
Problem Set 4

1. One solution to external fragmentaion is compaction. Compaction is the process of shuffling separated memory segments so that all free memoy is in one contiguous block.

Another solution is segmentation. Through segmentaion, memory is divided into variable sized chunks or segments. Logical addresses consist of a segment number and an offset. Although this does not completely solve the problem of segmentaion, it greatly reduces the amount of external fragmentation since memory can be broken up into smaller segments.

The third solution to external fragmentaion is paging. Through paging, physical memory is broken up into fixed size blocks called frames. Logical memory is broken into similar size blocks called pages. The backing store is also broken into fixed size blocks. Logical addresses consist of a page number and offset. A page table takes care of mapping logical addresses in pages to physical addresses in frames. Each process has its own page table and a pointer to that page table is stored in the PCB. This method solves the problem of external fragmentaion completely, but internal fragmentation can still occur.

2. a. 2200
b. 1000
c. 2200
d. 400
e. 2200
f. 2200

3. For this memory manegement system we can use paged page tables to organize pages. The page table for the page tables can contain three entries, two for process 1's page table and one for process 2's page table. The page table page table will reside at frame 0. At frames one and 2 will be the page table for process 1. This table will contain 7 entries. Process 1 resides at frames 4-10. At frame 3 will be the page table for process 2. This table contains 4 entries. Process two resides at frames 11-14.

Process 1 Page Table	Process 2 Page Table
0 | 4	0 | 11
1 | 5	1 | 12
2 | 6	2 | 13
3 | 7	3 | 14
4 | 8	
5 | 9	
6 | 10	

4. average memory access time = T * P_TLB + (1-P_TLB) * (M + P*D)
1ns * 0.9 + 0.1 * (10ns + 0.001*10e6) = 1001.9ns

5. LRU replacement does not suffer from Belady's anomoly because it is a stack algorithm. With a stack algorithm, the set of pages in memory for n frames is always a subset of the set of pages that would be in memory with n+1 frames. The set of pages in memory for LRU replacement is the n most recently used frames. Consider the following reference string with 3 frames available.
4 3 2 4 1 4. Using a FIFO page replacement algorithm, A page fault occurs three times. Using LRU only two page faults occur (excluding

the first three to fill the list).

6. Ref string: 3 2 4 3 4 2 2 3 4 5 6 7 7 6 5 4 5 6 7 2 1
FIFO:
memory: (3__), (32_), (324), (524), (564), (567), (467), (457), (456),
(756), (726), (721) = 12 Page faults
OPT:
memory: (3__), (32_), (324), (524), (564), (567), (564), (764), (724),
(721) = 10 page faults
LRU:
memory: (3__), (32_), (324), (524), (564), (567), (564), (567), (267),
(217) = 10 page faults
OPT and LRU generate the least amount of page faults.