Peter Woodman
0127119
CSE 410- H. Levy
Homework 6

1. External fragmentation describes fragmentation of physical memory between program spaces. Internal fragmentation refers to the space unnecessarily allocated to a program simply because of the memory block size.

2. (a) If we want to resize a process' address space, there either must be free memory aboufe the process in physical memory, or we must be able to move the process' memory space to another hole that can fit it. If the process is moved, it must somehow be re-linked. This may or may not work.
   (b) In this case, there either must be a hole above the process' physical address space, or the OS must be able to move the whole thing to another hole large enough to fit the process' new size. Icky!
   (c) In this case, there simply must be more pages available. That's it.

3. So that we can support a larger logical-address space more efficiently (i.e. without a huge page table)

4. (a) 649
   (b) 2310
   (c) trap. segfault.
   (d) 1727
   (e) trap. segfault.

5. Without virtual memory, a page fault means we've called a bad address. With virtual memory, it could also mean a page needs to be swapped in from the disk to a frame.

6. Copy-on-write is a feature that can be used when a process forks on a paging system. When the process forks, it initially shares some pages with the parent process, which are marked copy-on-write (the paging unit must understand this flag). When either process tries to modify something in one of these pages, the page is copied. Otherwise, it's not, and we save an unnecessary copy.

7. If there is a frequently used variable, and it is used every n+1 accesses to memory, and there are n frames available, LRU weill be worse than LFU. However, if there is a page that is used frequently in the initialization of the program and then never again, it will stay in memory far too long.

8. The machine looks up this address in the page table and finds the proper frame. It then accesses the frame with an offset of 2816. This is the proper location.

9. This means that 80 percent of the time, we have 1 μs access time, 18 percent of the time, we have 2 μs access time, and 2 precent of the time, we have horrible 20ms access time, assuming the 20ms figure refers to the time it takes to load a page to memory. Averaging this out, we have 401.16 μs access time on average.