Dave Sizer
CS370 A2

1.

- First fit:
  - Process E starts and requests 300 memory units.
    - 0-300 A
    - 300-600 E
    - 600-900 FREE
    - 900-1500 B
    - 1500-1800 FREE
    - 1800-2000 C
    - 2000-3000 FREE
    - 3000-3400 D
  - Process A requests 400 more memory units.
    - 0-300 FREE
    - 300-600 E
    - 600-900 FREE
    - 900-1500 B
    - 1500-1800 FREE
    - 1800-2000 C
    - 2000-2700 A
    - 2700-3000 FREE
    - 3000-3400 D
  - Process B exits.
  - Process F starts and requests 800 memory units.
    - 0-300 FREE
    - 300-600 E
    - 600-1400 F
    - 1400-1800 FREE
    - 1800-2000 C
    - 2000-2700 A
    - 2700-3000 FREE
    - 3000-3400 D
  - Process C exits.
  - Process G starts and requests 900 memory units
    - 0-300 FREE
    - 300-600 E
    - 600-1400 F
    - 1400-2100 A
    - 2100-3000 G
    - 3000-3400 D
- Best-fit
  - -
    - 0-300 A
    - 300-900 FREE

        - 900-1500 B
        - 1500-1800 E
        - 1800-2000 C
        - 2000-3000 FREE
        - 3000-3400 D
    - -
        - 0-700 A
        - 700-900 FREE
        - 900-1500 B
        - 1500-1800 E
        - 1800-2000 C
        - 2000-3000 FREE
        - 3000-3400 D
    - -
        - 0-700 A
        - 700-1500 F
        - 1500-1800 E
        - 1800-2000 C
        - 2000-3000 FREE
        - 3000-3400 D
    - -
        - 0-700 A
        - 700-1500 F
        - 1500-1800 E
        - 1800-2700 G
        - 2700-3000 FREE
        - 3000-3400 D
- Worst-fit
    - -
        - 0-300 A
        - 300-900 FREE
        - 900-1500 B
        - 1500-1800 FREE
        - 1800-2000 C
        - 2000-2300 E
        - 2300-3000 FREE
        - 3000-3400 D
    - -
        - 0-700 A
        - 700-900 FREE
        - 900-1500 B
        - 1500-1800 FREE
        - 1800-2000 C
        - 2000-2300 E
        - 2300-3000 FREE

        - 3000-3400 D
    - -
        - 0-700 A
        - 700-1500 F
        - 1500-1800 FREE
        - 1800-2000 C
        - 2000-2300 E
        - 2300-3000 FREE
        - 3000-3400 D
    - -
        - 0-700 A
        - 700-1500 F
        - 1500-1800 E
        - 1800-2700 G
        - 2700-3000 FREE
        - 3000-3400 D
- In this case best fit was the best because it required the least amount of compaction

2.

- There are 16 bits in both the virtual and physical address
- 6400 = (6x1024)+256

3.
A TLB with an entry for every frame with not be fully associative, so you can store more entries, but misses will be more common. With a smaller TLB, full associativity can be accomplished. Since misses from less-than-full associativity are far more expensive in OS operation than the slightly increased hit time of a fully associative cache, the smaller TLB is preferred.

4. Data sharing is easier in a paged memory scheme, because of the external fragmentation problem that segmentation presents. It is much easier for multiple processes to access a shared memory segment if it stays at the same location in memory. Under segmentation, as context switches take place, the private data for processes being swapped in and out of memory may cause the shared memory to be moved around to accommodate the swaps. In the case of paging, the hot-swappable constant size pages prevent this external fragmentation, making it easier to keep the shared memory in the same place. Even if it does change, the references to the frames in the page tables just need to be updated in each process.

5. One solution for processes that have more pages than the size of the table is to swap the page tables between primary and secondary memory, and the TLB. In the given example, you would split up the page entries into four chunks of 128 entries each. The four most significant bits of a virtual address would indicate which chunk of the page table the entry resides in. You could keep the most recently used chunk in the TLB, and the two most recently used chunks in the table in main memory. The two least used chunks could be in secondary memory (the disk). If we needed an entry that is not in the TLB, we would look to swap it from main memory. If it wasn't there, we would swap it from the disk. This would not change the virtual memory size, since the indicator of the table chunk is built into the

page table entry bits.  On average, the access time wouldn't be adversely affected because the principle of locality would state the most of the time programs would stay within the same page table segment, so swaps would not have to be frequent.

6.

- MIN
  - 0 - -
  - 0 1 -
  - 0 1 2
  - 3 1 2
  - 3 1 2
  - 3 1 4
  - 3 1 4
  - 3 1 4
  - 3 1 4
  - 3 5 4
  - 3 2 4
  - 3 2 4
  - 3 6 4
  - 3 6 4
  - 3 6 4
  - 3 6 4
  - 3 6 4
  - 3 6 4
  - 3 6 4
  - 3 6 7
  - 3 cold misses and 5 other misses
- FIFO
  - 0 - -
  - 0 1 -
  - 0 1 2
  - 3 1 2
  - 3 1 2
  - 3 4 2
  - 3 4 2
  - 3 4 1
  - 3 4 1
  - 5 4 1
  - 5 2 1
  - 5 2 4
  - 6 2 4
  - 6 3 4
  - 6 3 4
  - 6 3 4
  - 6 3 4

    - 6 3 4
    - 6 3 4
    - 6 3 4
    - 6 3 7
    - 3 cold misses and 9 other misses
- LRU
    - 0 - -
    - 0 1 -
    - 0 1 2
    - 3 1 2
    - 3 1 2
    - 3 4 2
    - 3 4 2
    - 3 4 1
    - 3 4 1
    - 3 5 1
    - 2 5 1
    - 2 5 4
    - 2 6 4
    - 3 6 4
    - 3 6 4
    - 3 6 4
    - 3 6 4
    - 3 6 4
    - 3 6 4
    - 3 7 4
    - 3 cold misses and 9 other misses.

7.

- Both the virtual and physical addresses would be 15 bits (10 for offset and 5 for page index).
- 10107 = (9 x 1024) + 891
- A reference to a byte in this page would result in a page fault.