Homework #3

Due in class on Friday 30 November 2001

1. **(Exercise 9.6 from OSC 6th edition, 8.6 in 5th edition) Consider a system where a program can be separated into two parts: code and data. The CPU knows whether it wants an instruction (instruction fetch) or data (data fetch or store). Therefore, two base-limit register pairs are provided: one for instructions and one for data. The instruction base-limit register air is automatically read-only, so programs can be shared among different users. Discuss the advantages and disadvantages of this scheme.**

    Advantages: Users can easily share code, meaning only one copy of that code would have to be in memory. The read-only segment would also have extra protection, since it would not only be protected against other processes accessing it, but it would be protected against this process accidentally trying to modify it. Also, if the one base-limit register scheme requires all the program to be stored contiguously, the two base-limit register scheme would at least allow the program to be stored in two separate partitions, giving the memory allocator more flexibility.

    Disadvantages: Extra hardware is required over the one base-limit register pair scheme — an extra set of registers, possibly (though not necessarily since it can be reused) extra comparison hardware. This method also has all the usual disadvantages of partitioning, but then again, so does the one base-limit register pair scheme.

2. **Consider the following page reference string: d c b a d c e d c b a e. If there are 3 frames in physical memory, how many pages faults will occur for (a) FIFO replacement, and (b) LRU replacement. Draw a diagram for each to show your work.**

    For three frames, FIFO = 9 page faults

    ```
    d, c, b, a, d, c, e, d, c, b, a, e
    d        a        e
      c          d                  b
        b            c                  a
    ```

    For three frames, LRU = 10 page faults

    ```
    d, c, b, a, d, c, e, d, c, b, a, e
    d        a        e        b
      c          d        x        a
        b            c        x        e
    ```

3. **(Exercise 11.4 from OSC 6th edition, 10.4 in 5th edition) Similarly, some systems support many types of structures for a file's data, while others simply support a stream of bytes. What are the advantages and disadvantages?**

    Many types of structures: The main advantage is increased ease of programming for the programmer. The main disadvantage is the increased complexity of the OS, which has to support, or at least recognize, all those file types.

    Stream of bytes: The main advantage is simplicity for the OS — it does not need to know how to handle different types of structures, and new file types can be supported easily without

modifying the OS. The main disadvantage is that the burden of interpreting the contents of the file now falls totally on the programmer; it can also be difficult for random access of data.

4. Compare a file system that supports indexed allocation to a file system that supports contiguous allocation, in terms of performance.

For sequential access, contiguous allocation results in better performance since little or no seeks are needed once file reading starts. Indexed allocation may happen to have its blocks stored contiguously, but contiguous storage is not guaranteed, so it may have worse performance for sequential files.

However, indexed allocation may have better performance for random access, particularly for a block near the end of the file, since it can go directly to the specific disk block instead of having to read all the previous blocks first.

5. (Exercise 14.10(a) from OSC 6th edition, 13.10(a) in 5th edition) Requests are not usually uniformly distributed. For example, a cylinder containing the file system FAT or inodes can be expected to be accessed more frequently than a cylinder that contains only files. Suppose that you know that 50 percent of the requests are for a small, fixed number of cylinders. Would any of the scheduling algorithms discussed in this chapter be particularly good for this case? Explain your answer.

SSTF would probably be best, as most of the time the head would stay within a small area. However, since most of the requests are occurring naturally within that area, then FCFS would also keep the head in the small area much of the time without the computational overhead of SSTF. One might also make a similar argument for LOOK and C-LOOK, though overall I think SSTF would be best.