Name ______________________ Student No. ______________________

No aids allowed. Answer all questions on test paper. Use backs of sheets if necessary.

Total Marks: 50

[10] 1. (a) Name two differences between logical and physical addresses.

–In the execution-time address-binding scheme, logical and physical-address spaces differ.

–The logical address is generated by the CPU, while the address seen by the memory unit is the physical address.

–The logical address is contiguous, the physical is may be spread out all over.

(b) Explain the difference between internal and external fragmentation.

–Internal Fragmentation is the area in a page that is not used by the job occupying that page.

–External Fragmentation exists when enough total memory space exists for a process, but is is not contiguous, so it cannot be assigned by one of the shcemes given in part (c) below.

(c) Describe briefly the following allocation algorithms: (i) First Fit, (ii) Best Fit, and (iii) Worst Fit.

(i) seach the list of available memory and allocate the first block that is big enough.

(ii) . . . smalles block that is big enough.

(iii) . . . largest block.

[15] 2. Consider the following request sequence:

$$\sigma = 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6$$

How many page faults would occur for the following replacement algorithms with a cache of size 3?

LRU replacement (15)
FIFO replacement (16)
OPT replacement (11)

[15] 3. Describe the OPT page replacement algorithm. Why is it considered to be infeasible? Show that OPT does not suffer of Belady's anomaly.

–OPT: replace the page whose next request is farthest in the future; if there are one (or more) pages that will never be requested again, evict any one of those.

–it is infeasible because in general we cannot predict the future.

–Let $\mathrm{OPT}_i(\sigma)$ =number of page-faults on a request sequnce σ with a cache of size i. We show that $\forall i, \sigma$, $\mathrm{OPT}_i(\sigma) \leq \mathrm{OPT}_{i+1}(\sigma)$. This is enough to show that OPT does not suffer of Belady's anomaly.

Run OPT_i and OPT_{i+1} side by side on the same request sequnce σ. If a page p is in the cache of OPT_i, it is also in the cache of OPT_{i+1}. To see this, suppose that for the first time in the request sequence σ, p is in the cache of OPT_i but not in the cache of OPT_{i+1}. Since OPT is a demand-paging algorithm, it means that p was requested in σ, and then evicted from OPT_{i+1} but not evicted from OPT_i. Hence, at some point p was the page in the cache of OPT_{i+1} whose next request was farthest in the future, and so it was evicted. But then it would have also been evicted from OPT_i. This is not possible, and so no such page p can ever exist.

From this it is clear that if OPT_{i+1} has a page-fault, OPT_i must also have a page fault.

[10] 4. (a) Why do some systems keep track of the type of a file, while others leave it to the user? Which system is "better"?

(b) 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?

–ANSWER TO BOTH QUESTIONS: Some systems allow different file operations based on the type of the file (for instance, an ascii file can be read as a stream while a dtabase file can be read via an index to a block). Other systems leave such interpretation of a file's data to the process and provide no help in accessing the data. The method which is "better" depends on the needs of the processes on the system, and the demands the users place on the OS. If a system runs mostly database applications, it may be more efficient for the OS to implement a databese type file and provide operationsl rather than making each program implement the same thing (possibly in different ways). For general purpose systems it may be better to only implement basic file types to keep the OS size smaller and allow maximum freedom to the processes on the system.