CS604 Operating systems

Final Term Examination – Spring 2005

Time Allowed: 90 Minutes

Please read the following instructions carefully before attempting any question:

1. **This examination is closed book, closed notes, closed neighbors.**
2. **Answer all questions.**
   a. **There is no choice.**
   b. **You will have to answer all questions correctly in this examination to get the maximum possible marks.**
3. **Do not ask any questions about the contents of this examination from anyone.**
   a. **If you think that there is something wrong with any of the questions, attempt it to the best of your understanding.**
   b. **If you believe that some essential piece of information is missing, make an appropriate assumption and use it to solve the problem.**
4. **Examination also consists of multiple-choice questions. Choose only one choice as your answer.**
   a. **If you believe that two (or more) of the choices are the correct ones for a particular question, choose the best one.**
   b. **On the other hand, if you believe that all of the choices provided for a particular question are the wrong ones, select the one that appears to you as being the least wrong.**
5. **All Programming questions should be answered using C syntax. Syntax errors will not be considered as errors. So try to only answer the question and put your idea and concept using C. Don't use any tool or IDE.**

****WARNING: Please note that Virtual University takes serious note of unfair means. Anyone found involved in cheating will get an `F` grade in this course.**

Total Marks: 50 **Total Questions: 12**

Question No. 1 **Marks : 04**

Assume that we have a demand-paged memory. The page table is held in CPU registers. It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds.
Assume that the page to be replaced is modified 70% of the time. What is the *maximum acceptable page-fault rate* for an effective access time of no more than 200 nanoseconds. Show your work.

Question No. 2 **Marks : 25**

Answer the following questions:

a. (4 Points) Write down names of two process related and two file related system calls in UNIX/Linux.

b. (2 Points) What does the following **region statement** mean?
region buffer when (count < n) do S1;

c. (4 Point) Consider three **cooperating processes** P1, P2, and P3. Write the code structures for the three processes using semaphores, insuring that statement S1 in P1 is not executed before statement S2 in P2, and S2 in P2 is not executed before S3 in P3. Show all your work, including initial values for the semaphores used in your code.

d. (2 Point) A system uses a CPU with a memory-to-memory move instruction whose operands may have up to two levels of indirection for both operands. What is the **minimum number of frames** that a well-behaved process must have in this system to guarantee its successful execution? Why? Assume that the system has 512 MB RAM, 100 GB hard disk, 4 KB pages, and 16-bit memory operands. Show your work.

e. (5 Point) List five basic **operating system components**.

f. (2 Point) Consider a **UNIX-based system** that uses a block size of 4 KB. If block addresses take 8 bytes, calculate the total **pointer overhead for the biggest possible file** on this system. An expression would suffice. What is the **size (in bytes) of the largest file** that can be stored on this system? Assume that the file inode contains 10 pointers to file blocks. Show all your work.

g. (1 Point) What is the purpose of **exponential averaging** technique?

h. (1 Point) What is **process address space**?

i. (1 Point) Name **standard files in UNIX/Linux**.

j. (3 Points) What is the purpose of medium-term scheduler? Give two reasons for its invocation (i.e., for it to be executed by the OS kernel).

Question No. 3 **Marks : 06**

Consider a file currently consisting of 100 blocks. Assume that the PCB, bit vector block, index block (in case of indexed allocation), and directory block are in the main memory. Calculate the *number of I/O operations* required to add a new block in the middle of the file for contiguous, linked, and indexed allocation techniques. For contiguous allocation, assume that there is no room to grow in the beginning or end of file. Also assume that block information to be added is in a kernel buffer. Show your work.

Contiguous Allocation:
Linked Allocation:
Indexed Allocation:

Question No. 4 **Marks : 05**

Consider a pure demand paging system with a paging disk that has an average access and transfer time of 20 milliseconds. Addresses are translated through a page table in main memory, with an access time of 50 nanoseconds per memory access. Thus, each memory reference through the page table takes two accesses. To improve this time, we have added cache memory that reduces access time to one memory reference, if page table entry is in the cache. Assume that cache access time is negligible.
Assume that 80% of the accesses are in the cache, of the remaining, 10% cause page faults, and 50% of the replaced pages are dirty. Write down the expression / equation for the *effective memory access time* for the computer system. Show your work.

Question No. 5 **Marks : 20**

Consider a computer system with the following specifications and answer the questions that follow. The system uses a cache (TLB) to contain part of the page table for a process. Show your work.

a. (4 Points) What are the sizes of **physical address** and **physical address space**?
Physical Address:
Physical Address Space:

b. (4 Points) What are the sizes of **logical address** and **logical address space**?
Logical Address:
Logical Address Space:

c. (2 Point) What is the size (in bytes) of the **page table**?

d. (3 Points) How many **levels of paging** does the system support? Why? Show the break down of the logical address in the form of $\mathbf{p_i}$'s and $\mathbf{d}$. Clearly show the number of bits needed for $\mathbf{d}$ and each $\mathbf{p_i}$.

e. (3 Point) If page 7 of a process is stored in frame 126 and the process generates the logical address (7, 16), show **logical** and **physical addresses** in the form of 1s and 0s?

f. (4 Points) What is the **effective memory access time** for the system? Show your work.

Question No. 6 **Marks : 10**

Consider the following snapshot of a system of 5 processes, P0 through P4 and four resource types, A, B, C, and D. Use the *banker's algorithm* to determine if the system is in a *safe state* by providing a *safe sequence*. If the system is not in a safe state, describe why it is so. Show all your work.

Question No. 7 **Marks : 10**

The following is a solution for the *Bounded-Buffer Problem* by using semaphores. Is the solution correct? Why? If not, point out errors, explain why they are errors, and correct them.

```
semaphore empty = 0;
semaphore full  = N;
semaphore mutex = 1;
```

Question No. 8 **Marks : 04**

A system call

1. *Is an entry point into the kernel code*
2. *Allows a program to request a kernel service*
3. *Is a technique to protect I/O devices and other system resources*
4. *All of the above*
5. *None of the above*

Question No. 9 **Marks : 04**

A system uses a page size of 8 KB and has 512 MB RAM. If it allows a maximum process size to be 64MB, then the following is true for the system.

*[Note that **B** in KB, MB, and GB stands for "**Bytes**" and NOT "Bits".]*

1. *It has 29-bit logical address*
2. *It has 26-bit physical address*

3. *It has 64 K frames*
4. *All of the above*
5. *None of the above*

Question No. 10 **Marks : 04**

Semaphore operations, `wait` and `signal` , must be performed atomically because:

1. *If they are not atomic, the semaphore value may become incorrect.*
2. *They are implemented as system calls.*
3. *Otherwise busy waiting becomes a problem due to the wasted CPU cycles.*
4. *The* **`wait`** *operation increments the value of the semaphore by one and signal decrements it by one, we must not allow interleaving of these operations as the semaphore value may become incorrect.*
5. *None of the above*

Question No. 11 **Marks : 04**

A page replacement algorithm in a demand-paged virtual memory system may suffer from Belady's Anomaly if

1. *The number of frames allocated to a processes are increased from three to four*
2. *The replacement algorithm does not belong to the class of stack replacement algorithms*
3. *The replacement algorithm is not the Least Recently Used algorithm*
4. *All of the above*
5. *None of the above*

Question No. 12 **Marks : 04**

When a soft link is created in UNIX with the `ln –s` command:

1. *A new directory entry is created*
2. *A new file is created that contains the pathname of the existing file as specified in the command.*
3. *The inode numbers in the directory entry for the existing file and that of the link file are different.*
4. *All of the above*
5. *None of the above*