**The Universal Operating System**

**MEMORY**

1) On this page table, What does it mean if the dirty bit is set.  (Circle all that apply)

 ◦ The page is not in RAM but swapped out to disk.

 ◦ The page is shared among several processes.

 ◦ The process does not have permission to write to that page .

 ◦ The page has been changed since it was read from the disk.

 ◦ None of these silly answers

2) Suppose the Fjord-making machine runs the following code.  Remember that Fjord making computer uses a two-level page table.  Is the address a virtual or physical address?

 setToTwelve() {

   char \*depth = 0123456;

   \*depth = 12;

 }

3) Normally, in real life, what kind of page does an O.S. swap out when it needs to free RAM?

 a recently used page

 a soon to be used page

 a page used in the distant past

 a page who's next use is the distant future

4) Using the page table below, where in physical RAM is virtual address 0?  Assume 1 kilobyte pages.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Physical Page Number | 2 | 3 | 4 | 7 | 6 |
| Dirty | N | N | N | Y | Y |
| Referenced | Y | Y | Y | N | N |

5) Using the page table above, what is the translation of physical address 0 into virtual addresses?

6) When do page faults occur? (circle all that apply, or zero)

 ◦ They can happen any time a process accesses it’s stack.

 ◦ The can happen any time a processes accesses it’s executable code.

 ◦ The can happen any time a process allocates ram.

 ◦ They can happen any time a process reads a global variable.

8) The memory subsystem of the coral reef planning computer uses 8K pages. If it must allocate 1000 memory allocations, averaging 100K each randomly distributed, and the computer uses 64 bit pointers and a 3 level page table scheme, what’s the average internal fragmentation for this computer?

**DISKS**

1) There is a machine that makes rocks. It’s very important to keep an inventory of all rocks ever made. This database needs to hold 70G of data. Using RAID 5 and 10G disks, how many disks would be needed to hold the data and survive loss of a disk.

2) Suppose they decide to run with RAID 1 (mirroring). Nothing else changes. How many disks are needed?

3) Using RAID 5 and 10 disks including the 1 parity disk and each disk can do 100 read accesses per second, how many read accesses per second can the whole array do?

4) What is the mechanism for storing data in a solid state SSD?

 A magnetized region

 Trapped electrons

5) Normal spinning metal disks work best at

 reading a very few super long files

 reading a zillion super small files

 **FILE SYSTEMS**

1) I want a file system that can quickly seek to different parts of the same file, and do so even if the no part of the file system is cached. Which file system should I use?

 VFAT EXT2

2) Beyond the file name and extension, list something stored in a VFAT directory

3) Beyond the file name and extension, list something stored in an EXT2 directory

**SCHEDULING**

1) Which takes longer, to create a thread or to create a process?

2) Beyond the time delay, what does a process notice when the scheduler takes the CPU and then later gives it back? Choose zero or more.
 a) The CPU registers have changed.
 b) The stack has changed.
 c) The memory for this process has changed.

3) Who does scheduler aging help? Choose exactly one.
 a) Low/Weak priority processes
 b) High/Strong priority processes

4) Which of the following algorithms is subject to starvation? Assume no program has an infinite loop. (choose zero or more).

* + - First Come First Serve
		- Round robin
		- Shortest Job Next
		- Highest Priority Next
		- Multilevel feedback queue without aging

5) Suppose you have the following processes to run under priority scheduling without preemption.  Which process is running at time slice #10?  Low numbers are low/weak priorities. A process that arrives at time 1 can run at time 1.

      Job          Priority         Arrival Time    Length

      ---------------------------

        1                10                     1                       5

        2                 5                      2                       3

        3                 12                    3                       17

        4                  6                     4                       8

6) Oops .. it was FCFS. What’s run at time 23?

7) What is the difference between hard real time and soft real time?

**MULTIPROCESSING**

1) Which of the following things are shared by any two threads that came from the same process? Choose zero or more.
 Code File Descriptors Page Table
 Data from new/malloc C++ local variables c++ global variables

2) (Yes/No) The operating system for the continental drift computer uses the following rules to coordinate its locking scheme. All locks must be asked for in alphabetic order (disk before printer before soundcard, for example). However anyone may request the page table lock at any time. Given these to rules, is deadlock possible?

 3) What is the most COMMON reason  we want to overlap computation and I/O?

 ◦ otherwise operating system class would be boring.

 ◦ DOS does not, so there is no need to.

 ◦ Some common computations are not possible without overlapping computation and I/O.

 ◦ Total system throughput can be faster with overlapped computation and I/O.

4) In my system, some shared global variable is accessed in only read only mode by every thread. Who must lock?

 All the threads At least one thread None of the threads

**MISC**

1) The biggest supercomputers use how many CPUs?

 tens thousands millions billions

2) What does exec/execv/execvp return if it succeeds?

3) (Local/Global) In the code below, can mtx be declared as a local variable or must it be a global variable?

void print\_block (int n, char c) {

 mtx.lock();

 for (int i=0; i<n; ++i) {

 std::cout << c;

 printf("%c", c);

 }

 printf("%c", c);

 mtx.unlock();

}