CS 241, Homework #2, Fall 2009

Name: _____ (*and place your NetID on every sheet!*)

INSTRUCTIONS:

- This homework is **INDIVIDUAL WORK**. This homework **SHOULD NOT** be done with your MP team.
- Please **TYPE** your homework solutions. Handwritten solutions **WILL NOT** be graded.
- Please **STAPLE** all sheets together. And type your NetID on **EVERY** stapled sheet.
- Homework is due **IN-CLASS, AT THE BEGINNING OF CLASS (11:00am)** on Monday, December 7, 2009. Since solutions will be discussed during class, no late submissions are allowed.

GRADING RUBRIC:

	Multiple Choice /10	Fill In The Blank /11	Missing Code /6	Total /27
Grade:				
Grader:				

---

1. Two schemes are commonly used to manage free spaces for memory dynamic partitions: bit map and linked list. Which of the following is correct about these two schemes:
   A. In most cases, linked list is more space efficient
   B. In terms of processing time (updating the data structure due to allocation and deallocation), linked list is faster on average.
   C. On average, linked list is faster to find a free hole.
   D. A and B
   E. A and C
   F. All of the above

2. Which one of the following best describes the Translation Lookaside Buffer?
   A. Is used to convert a physical address into a page offset.
   B. Is used to convert a virtual address into a page offset.
   C. Is used to cache the results of recent virtual page requests.
   D. Is used for asynchronous virtual memory updates in concurrent systems.
   E. Is used to create process queues in a public virtual address space.

3. A server is 95% utilized. Assuming a single queue, Poisson arrivals, and exponentially distributed service times, if the average service rate is 10 requests/second, what is the average queuing time for an incoming request?
   A. 0.1 seconds
   B. 0.95 seconds
   C. 1.8 seconds
   D. 1.9 seconds
   E. 2.0 seconds

4. If the incoming request rate on the above server dropped until its average utilization became 90%, what is the average number of requests in the entire system?
    A. 0.9 requests
    B. 8.1 requests
    C. 9.0 requests
    D. 10.0 requests
    E. None of the above

5. Which of the following is the correct order of (a subset of) calls for a *TCP client*?
    A. socket(); bind(); listen(); accept();
    B. bind(); socket(); listen(); accept();
    C. bind(); socket(); recvfrom();
    D. socket(); bind(); recvfrom();
    E. socket(); connect();

6. Which one of the following best describes disk I/O using polling?
    A. CPU time is required to continually check the status of I/O hardware and read/write bytes as needed
    B. The CPU is interrupted when the I/O device is ready to transfer bytes
    C. The I/O device transfers bytes to main memory without CPU intervention.
    D. The I/O device uses a poll channel to halt the current CPU instruction.
    E. The system bus is cross-linked to the memory bus during I/O transfer.

7. What information is **not** contained in a file's i-node in an i-node based file-system?
    A. The size of the file
    B. The last modification time of the file
    C. The name of the file
    D. The permissions of the file
    E. None of the above (all data listed in the answers above is contained in the i-node)

8. Which of the following file descriptor statements is **not** correct?
    A. File descriptors are local to each process.
    B. When a process is forked, its children can inherit open file descriptors.
    C. Each process can have several open file descriptors.
    D. Different processes can have file descriptors addressing the same physical file
    E. Each thread has its private set of open file descriptors

9. Consider a series of disk cylinder requests that arrive to the disk controller at exactly the same time: 50, 81, 55, 20, 120, 65, 35. If the disk head is currently on cylinder 61, which request is served last using Shortest Seek Time First?
    A. 120
    B. 81
    C. 65
    D. 20

10. Assuming a page size of 4096 bytes and that a page table entry takes 4 bytes, how many levels of page table would be required to map a 42-bit address space (byte-addressable) if the top level page table fits into a single page?
    A. 1
    B. 2
    C. 3
    D. 4

---

11. Suppose your machine has 32 bit addresses and uses a 2-level page table. Virtual addresses are split into: a 8 bit top-level page table field, an 12 bit second level page table field, and an offset.
    A. **(1 point)** How large are the pages?

    ANSWER: _____ bytes

    B. **(1 point)** How many pages are there in the address space?

    ANSWER: _____ pages

12. Consider the following page table below. All numbers are decimal, everything is numbered starting from zero, and all addresses are memory byte addresses. The page size is 1024 bytes and the processor architecture is 32-bit.

Virtual Page #	Valid Bit	Dirty Bit	Page frame #
0	1	0	4
1	1	1	7
2	0	0	-
3	1	0	2
4	0	0	-
5	1	1	0

What physical address, if any, would each of the following virtual addresses correspond to? (Do not try to handle any page faults if any). Note that the addresses are decimal addresses, not hexadecimal addresses

    A. **(1 point)** Virtual Memory Address: 1052 (decimal)

    ANSWER: _____ (decimal)

    B. **(1 point)** Virtual Memory Address: 2221 (decimal)

    ANSWER: _____ (decimal)

    C. **(1 point)** Virtual Memory Address: 5499 (decimal)

    ANSWER: _____ (decimal)

13. **(1 point)** In some machine architecture, the TLB access time is 1 clock cycle and the memory access time is 10 clock cycles. What's the minimum TLB hit ratio for TLB to be worthwhile? You can assume that the page table needs a single memory access.

    ANSWER: _____

14. Consider a system with 4 page frames. Given the virtual memory reference string 0, 1, 7, 2, 3, 2, 7, 1, 0, 3 how many page faults occur if FIFO page replacement algorithm is used? How about LRU? Your answer should show the final content of the page frames after the 10 page references and the number of page faults.
    A. **(1 point)** First In First Out (FIFO) Page Replacement

       ANSWER: _____ page faults

       ANSWER: [0]=____ [1]=_____[2]=_____ [3]=_____

    B. **(1 point)** Least Recently Used (LRU) Page Replacement

       ANSWER: _____ page faults

       ANSWER: [0]=____ [1]=_____[2]=_____ [3]=_____

15. An inode-based file system is mounted on a disk with 4Kbyte blocks. Block addresses (also called "block pointers") are 32 bits. A disk inode contains 10 direct entries, a single indirect, a double indirect and a triple indirect entry. Single indirection is used once all ten direct entries are filled. What is the largest file (in Kbytes) that can be stored in the following situations:
    A. **(1 point)** When half of the direct entries are utilized?

       ANSWER: _____ Kbytes

    B. **(1 point)** When half of the entries in the single indirect block are utilized?

       ANSWER: _____ Kbytes

    C. **(1 point)** When the second indirect block is half full?

       ANSWER: _____ Kbytes

16. Consider the following program displayed below. There are six lines that have been marked to be filled in with code that we provide (on the next page). It may not be necessary to fill each line. If you complete all that is required of the code before filling all the provided lines for that chunk of code, simply choose the last option ("/* No coded needed on this line. */"). DO NOT provide your own code.

```
int main(int argc, char **argv)
{
   sigset_t signal_set;
   timer_t timer;
   struct itimerspec timer_time;
   int signal_received;
   float period;

   // Create a timer
   timer_create(CLOCK_REALTIME, NULL, &timer);

   // Compute the timer's period.
   period = 1/((float) 20);
   timer_time.it_value.tv_sec = (time_t) period;
   timer_time.it_value.tv_nsec = (long) ((period - (long) period)*1e9);
   timer_time.it_interval.tv_sec = timer_time.it_value.tv_sec;
   timer_time.it_interval.tv_nsec = timer_time.it_value.tv_nsec;

   // Set the timer's period.
   timer_settime(timer, 0, &timer_time, NULL);

   // Setup the process mask and signal set for sigwait().
   (1) _____________________________________
   (2) _____________________________________
   (3) _____________________________________

   while(1)
   {
      // Synchronously wait for the appropriate signal (the timer).
      (4) _____________________________________
      (5) _____________________________________

      // Act if the timer's signal arrives
      if(signal_received == (6) __________)
          do_io(); // DO SOME I/O - DETAILS REMOVED FOR BREVITY
   }
}
```

(Answers appear on the next page.)

(Question begins on the previous page.)

Please use one of these options to fill in each of the six lines marked in the code above. Please note that is may be necessary to use one or more of the lines for multiple answers. If you have completed all tasks necessary in the code and have lines remaining, please use the last answer to denote that no more functions are required.

A. sigaddset(&signal_set, SIGALRM);
B. sigaddset(&signal_set, SIGINT);
C. sigaddset(&signal_set, SIGUSR1);
D. sigaddset(&signal_set, NULL);
E. sigdelset(&signal_set, SIGALRM);
F. sigdelset(&signal_set, SIGINT);
G. sigdelset(&signal_set, SIGUSR1);
H. sigdelset(&signal_set, NULL);
I. sigemptyset(&signal_set);
J. sigfillset(&signal_set);
K. sigprocmask(SIG_BLOCK, &signal_set, NULL);
L. sigprocmask(SIGALRM, &signal_set, NULL);
M. sigprocmask(SIGINT, &signal_set, NULL);
N. sigprocmask(SIGUSR1, &signal_set, NULL);
O. sigwait(&signal_set, &signal_received);
P. sigwait(&signal_set, NULL);
Q. SIGALRM
R. SIGINT
S. SIGUSR1
T. NULL
U. /* No coded needed on this line. */

(6 x 1point) Provide your answer below (just supplying the letter corresponding with the answer is enough):

ANSWER Line (1): _____

ANSWER Line (2): _____

ANSWER Line (3): _____

ANSWER Line (4): _____

ANSWER Line (5): _____

ANSWER Line (6): _____