Department of Computer Science, National Tsing Hua University
Ph.D. Qualification Examination
Operating Systems, Spring 2017

1. (10%) Discuss the advantages and disadvantages of allocating a swap space as a large file in a normal file system and as a separate disk partition.

2. (10%) Describe three major disadvantages of linked allocation of disk space to files.

3. (10%) Describe how direct memory access (DMA) works.

4. (15%) Given a byte-address machine which has 4GB physical memory, 1KB page, 24bit virtual address, and physical addresses are stored in page table, answer the following questions:
   (a). (3%) How many bits for the physical address?
   (b). (3%) What is the maximum number of pages that can be allocated by a program?
   (c). (3%) If we use hierarchical page table structure to ensure page table size does not exceed the page size, what is the minimum levels of page table do we need?
   (d). (6%) Give a step-by-step example to illustrate how a virtual address can be translated into a physical address on this machine. (Your example must not contradict to the machine description.)

5. (10%) Consider a system that allocates pages of different sizes to its processes.
   (a). (4%) Give two advantages of such a paging scheme.
   (b). (6%) Discuss what modifications are required to provide this functionality in terms of the design of page table, TLB, and swap space.

6. (10%) Answer the following questions about interrupts:
   (a). (3%) What is the purpose of interrupts?
   (b). (3%) What are the differences between a trap and an interrupt?
   (c). (4%) Can traps be generated intentionally by a user program? If so, for what purpose?

7. (15%) The turnaround time of a job/process is defined as the interval from the time of submission of the job/process to the time of completion.

(a). (7%) Suppose that five jobs which arrive at the same time are waiting to be run in a **non-preemptive** system. Assume that each context switch takes 1 time unit. The expected CPU burst times of the five jobs are 8, 5, 2, 4, and *X*. In what order should they be run to minimize average turnaround time? The answer should depend on *X*.

(b). (8%) Suppose that *n* jobs which arrive at the same time are waiting to be run in a system which allows **preemption** of jobs. Assume that each context switch takes no time. The expected CPU burst times of the *n* jobs are given and denoted by t_1, t_2, ..., t_n. What scheduling algorithm can minimize the sum of averaged turnaround times? Prove that this scheduling algorithm (the one you answer) minimizes the sum of averaged turnaround time.

8. (10%) Consider 4 philosophers who spend their lives thinking and eating. The philosophers share a table on which there are 9 chopsticks. From time to time, a philosopher gets hungry and tries to use **3** chopsticks to eat. Suppose that a philosopher can pick up only one chopstick at a time. (Obviously, she cannot pick up a chopstick that is already in the hand of another philosopher.) When a hungry philosopher has **3** chopsticks at the same time, she eats without releasing her chopsticks. When she is finished eating, she puts down all of her chopsticks and starts thinking again. Prove or disprove that any of the philosophers may get starved.

9. (10%) Consider the below program.

(a). (5%) Including the initial parent process, how many processes are created by the program? You have to justify your answer.

```
#include <stdio.h>
#include <unistd.h>

int value = 100;

int main()
{
  do {

    pid_t pid = fork();  /* fork a child process */

    if (pid == 0) {    /* child process */
      // Many irrelevant lines here are removed to save space.
      value--;
    }
    else {    /* parent process */
      // Many irrelevant lines here are removed to save space.
      fork();
      execlp("\bin\ls");
      /* the function execlp() replaces the current process image
```

```
        with a
                new process image specified by file.  */
        }

    } while(value >= 0);

    return 0;
}
```

(b). (5%) Assume that the system is overloaded by the above program with such a number of processes. An idea to reduce the load is replacing processes by threads. Explain why the use of threads (rather than processes) can reduce the load.