國立彰化師範大學 101 學年度碩士班招生考試試題

系所： 資訊工程學系　　科目： 作業系統及計算機組織

☆☆請在答案卷上作答☆☆　　

一、 Using the program in Fig. 1, identify the values of ***pid*** at lines A, B, C, and D. (Assume that the actual ***pid***s of the parent and child are 1000 and 1002, respectively.) **(12%)**

```
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>

int main()
{
pid_t pid, pid1;

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

   if (pid < 0) { /* error occurred */
     fprintf(stderr, "Fork Failed");
     return 1;
   }
   else if (pid == 0) { /* child process */
      pid1 = getpid();
      printf("child: pid = %d", pid); /* A */
      printf("child: pid1 = %d", pid1);    /* B */
   }
   else { /* parent process */
      pid1 = getpid();
      printf("parent: pid = %d", pid); /* C */
      printf("parent: pid1 = %d", pid1);    /* D */
      wait(NULL);

   }

   return 0;

}
```

Fig. 1. What are the pid values?

二、 What are the advantages of the variant of linked allocation that uses a file-allocation table (FAT) to chain together the blocks of a file? **(10%)**

三、 When a page fault occurs, the process requesting the page must block while waiting for the page to be brought from disk into physical memory. Assume that there exists a process with five user-level threads and that the mapping of user threads to kernel threads is many to one. If one user thread incurs a page fault while accessing its stack, would the other user threads belonging to the same process also be affected by the page fault — that is, would they also have to wait for the faulting page to be brought into memory? Explain your answer. **(10%)**

國立彰化師範大學 101 學年度碩士班招生考試試題

系所： 資訊工程學系　　科目： 作業系統及計算機組織

☆☆請在答案卷上作答☆☆　　

四、 Which of the following CPU scheduling algorithms could result in starvation? **(6%)**

a. First-come, first-served
b. Shortest job first
c. Round robin
d. Priority

五、 Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free. **(12%)**

六、 Answer the following questions.

(a) In evaluating the performance of a computer, two performance metrics *response time* and *throughput* are usually used. Explain the meanings of these two terms. **(5%)**.

(b) *Amdahl's law* is a rule stating that the performance enhancement possible with a given improvement is limited by the amount that the improved feature is used. Give a computer-related example to explain this rule. **(5%)**

(c) Many programs have more variables than computers have registers. How does the compiler do to conquer this problem? **(5%)**

(d) In the execution of a procedure, the program must first put the parameters in a place where the procedure can access them, and (after performing the desired task) return control to the point of origin. How can these two steps be achieved in computer hardware for a MIPS machine? **(5%)**

(e) A designer of a floating-point representation must find a compromise between the size of the fraction and the size of the exponent. The tradeoff is between precision and range. What does that mean? **(5%)**

國立彰化師範大學 101 學年度碩士班招生考試試題

系所： 資訊工程學系　　科目： 作業系統及計算機組織

☆☆請在答案卷上作答☆☆　　

(f) Shown in Fig. 2 is a portion of the datapath for MIPS hardware, what is the purpose for using an adder here and why the value to be added to the PC is 4? **(5%)**

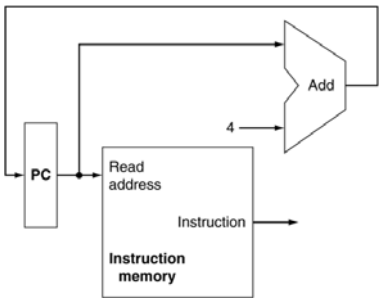

Fig. 2

(g) When an exception occurs, the address of the offending instruction will be saved in the exception program counter (EPC), and the control will be transferred to the OS. After performing some required actions, what will then be done by the OS? **(5%)**

(h) In a pipelined design with *N* stages, what is its theoretical speedup and how to determine its clock rate? **(5%)**

(i) When the two instructions are executed in the pipelined design in Fig. 3, what kind of hazard will occur? Why? **(5%)**

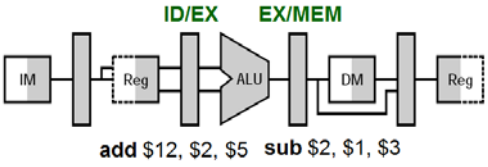

Fig. 3

(j) Suppose on a store instruction, the data is written only into the cache (without changing main memory), then the memory would have a different value from that in the cache. Two ways to solve this are *write-through* and *write-back*. Please explain their meanings. **(5%)**