請使用答案卡作答

一、複選題(80%)，共二十題，每題全對得4分，答對一個選項得1分(答對兩個選項得2分，以此類推)，答錯一個選項倒扣2分(答錯兩個選項倒扣4分，最多扣至該題0分為止)，整題未作答不給分。(舉例每題有a、b、c、d四個選項，某題答案為a、b，而作答時選a、b、c，則三個選項正確一個錯誤，該題得3-2=1分)

1. Which one(s) of the following statements about processes and threads are correct?
   (a) All the data in one thread are shared with other threads in the same process.
   (b) Kernel threads in the same process share one program counter.
   (c) When multiple user-level threads are mapped to a single kernel thread through a thread library, these user-level threads cannot take advantage of the multi-core structure.
   (d) Race condition can occur in a process in which multiple user threads are associated with a kernel thread.

2. Which one(s) of the following statements about modern operating systems are correct?
   (a) When a process writes a character on screen, this operation is accomplished with one or more system calls.
   (b) In a time-sharing system, a process will not leave the running state unless it terminates or it is preempted through a timer interrupt.
   (c) Applying the microkernel structure can reduce the communication overhead between kernel and user space.
   (d) One process can communicate with another process through shared memory.

3. Assume that a system issues multiple readers and writers processes with the following modified readers-writers pseudo codes:

```
Shared semaphore mutex, wrt initialized to 1.
Shared Integer readcount initialized to 0.
A writer process:
      wait (wrt) ;
      // writing the content
      signal (wrt) ;

A reader process:
      wait (mutex) ;
      readcount ++ ;
      if (readercount <= Na)  wait (wrt) ;
      signal (mutex)
      // reading the content
      wait (mutex) ;
      readcount-- ;
      if (readcount <= Nb)  signal (wrt) ;
      signal (mutex) ;
```

   Which of the following statements are correct?
   (a) When **Na**==1 and **Nb**==0, at most one process can access (read or write) the content at any time.
   (b) When **Na**==1 and **Nb**==0, at most one reader process is waiting for **wrt** at any time.

(c) When **Na**==2 and **Nb** == 1, at most two reader processes are waiting for **wrt** at any time.
(d) None of the above statement is correct.

4. If a system is monitored by the banker's algorithm and is currently in a safe state, which of the following changes can be made safely (without introducing the possibility of deadlock?)
(a) Increase the amount of available resources
(b) Increase the number of resource types
(c) Increase the Max for one process.
(d) Release the allocated but no longer used resources of one process.

5. Given the following codes with fork(),

```
#include <stdio.h>
#include <unistd.h>
void main()
{
    pid_t pid;
    int a;
    int i;
    a = 5;
    for(i=0; i<2; i++ ) {
        pid = fork();
        if(pid<0) {
        return;
        } else if(pid==0) {
            a-=2;
        printf("%d\n", a);
        } else {
            a--;
            printf("%d\n", a);
        }
    }
}
```

Which of the following statements are correct?
(a) The sum of all output numbers is larger than 14.
(b) At least one of the output numbers is smaller than zero.
(c) Besides the primary process, at most two additional processes are created.
(d) None of the above statement is correct.

6. Which one(s) of the following statements regarding virtual memory are true?
(a) Logical memory space can be much larger than physical memory space.
(b) Swapped-out pages are stored in a disk partition or a pre-allocated system file.
(c) When a page is replaced, the page must be written back to the swap space regardless of whether it has been modified or not.
(d) When a page fault occurs, the running process is trapped into the operating system kernel.

7. Consider three typical disk scheduling algorithms: First-Come First-Serve (FCFS), Short-Seek-Time First (SSTF), and SCAN (the elevator algorithm). Which one(s) of the following statements are true?
(a) FCFS never starves a disk I/O request.
(b) The variation of disk I/O latencies under SSTF can be very high.
(c) SCAN is optimal in terms of total seek distance.

(d) Disk scheduling significantly improves disk I/O performance on conventional hard drives, but not on RAM disks.

8. Consider the following directory listing information, which is produced using an "ls -l" command in a Linux console. The fields of each line stand for (in order): access permission, hard link count, owner name, owner group, file size in bytes, time stamp, and file name. Which one(s) of the following statements are true?

```
-rw-rw-r--  1 andy  staff      31219   Sep 3 08:30 intro.ps
drwx------  5 andy  staff      512     Jul 8 09.33 private/
drwxrwxr-x  2 andy  staff      512     Jul 8 09:35 doc/
drwxrwx---  2 andy  student    512     Aug 3 14:13 student-proj/
-rw-r--r--  1 andy  staff      9423    Feb 24 20:03 program.c
-rwxr-xr-x  1 andy  staff      20471   Feb 24 20:03 program
drwx--x--x  4 andy  faculty    512     Jul 31 10:31 lib/
drwx------  3 andy  staff      1024    Aug 29 06:52 mail/
drwxrwxrwx  3 andy  staff      512     Jul 8 09:35  test/
lrwxrwxrwx  1 andy  staff      9       Dec 21 16:51 t -> program.c
```

(a) There are 4 sub-directories in the directory "lib".
(b) The file "t" is a hard link.
(c) The file "program" can be executed by users in the group "staff".
(d) The amount of storage space allocated to file "intro.ps" is exactly 31,219 bytes.

9. Regarding page table designs, which one(s) of the following statements are true?
(a) In inverted page tables, a table entry contains a page number and a process ID.
(b) In hash-based page tables, if a hash bucket is associated with a linked list for collision handling, each element of the linked list contains a frame number plus a page number.
(c) In multilevel page tables, an inner page table entry contains a frame number.
(d) With multilevel page tables, the more levels a page table has, the higher the TLB miss penalty will be.

10. Consider three typical file allocation schemes: contiguous allocation, linked (link-list) allocation, indexed allocation, and extent allocation. Which one(s) of the following statements are true?
(a) Contiguous allocation offers the best read and write performance for large files.
(b) Files are prone to file fragmentation with the linked allocation scheme.
(c) Random file access with indexed allocation in general is faster than with linked allocation.
(d) File defragmentation is not necessary for extent allocation.

11. Consider the CPU performance, which of the following statements are true?
(a) If you want to decrease the response time of a task, you have to promote your PC by increasing the number of CPUs of your PC. Assume the old CPU and new CPUs in your PC are of the same type and have the same specification.
(b) If we run two programs on two CPUs of different ISAs to perform the same task, the CPU with smaller CPI must have less CPU time than the other CPU.
(c) The CPU time of performing a program on a CPU is determined by the instruction count, the CPI, and the clock rate of the CPU.
(d) Using Millions of Instructions Per Second (MIPS) as a metric to measure the CPU performance is not fair since MIPS does not consider the differences in the

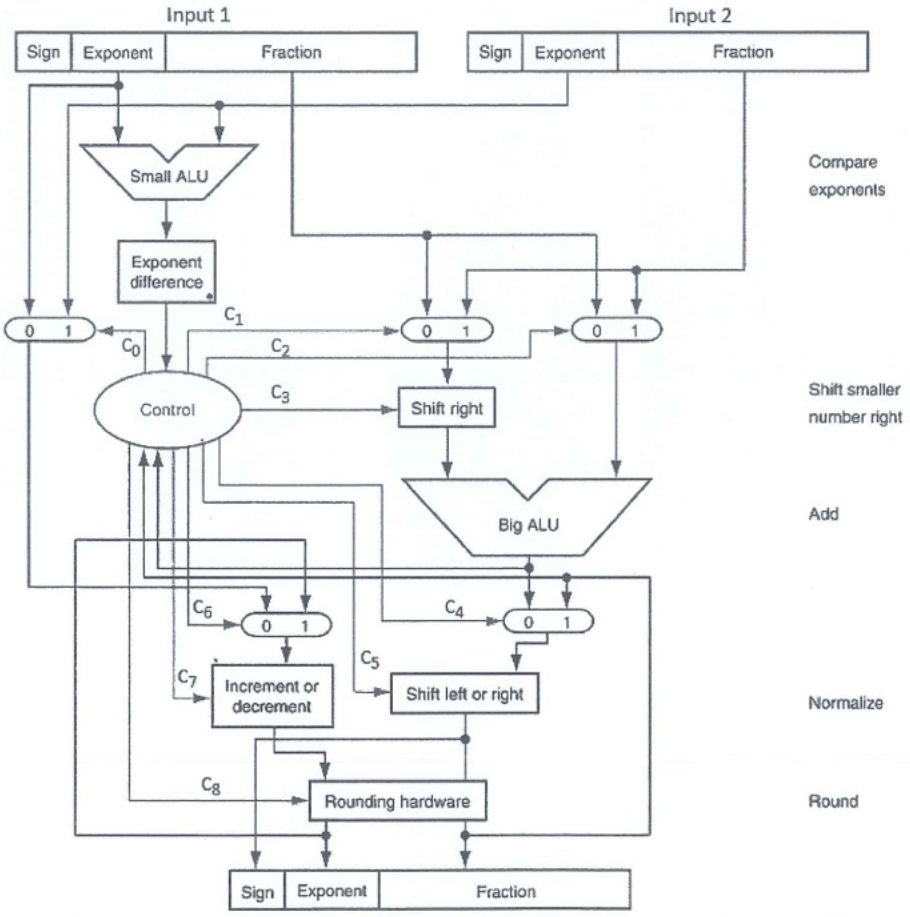
complexity among instructions.

12. Consider the design principles of MIPS CPU, which of the following statements are true?
    (a) *add* instruction has a fixed number of operands, i.e., 3, and does not have more operands because simplicity favors regularity, which also enables higher performance at lower cost.
    (b) MIPS has a small group of registers rather than a large and complex group of registers to form a register file, which shows the principle of favoring regularity by simplicity.
    (c) The immediate operand in immediate arithmetic instructions follows the principle of making the common case fast.
    (d) Several types of instruction formats in MIPS reveals the principle of the demands of compromises for good design.

13. Consider computer arithmetic and CPU design, which of the following statements are true?
    (a) For floating-point numbers, the sticky bit is used to promote the accuracy of rounding a floating-point number, where increasing the number of used sticky bits can improve the accuracy.
    (b) The merit of non-restoring division algorithm (NRDA) over restoring division algorithm (RDA) is that the NRDA requires one less addition than the RDA if the subtraction of the divisor register from the remainder register (dividend) yields a negative number.
    (c) The addition of two integers (represented by 2's complement) of different signs does not produces ah overflow.
    (d) The following codes perform the overflow check for the addition of two unsigned integers stored in $t1 and $t2. (addu $t0, $t1, $t2; nor $t3, $t1, $zero; sltu $t3, $t3, $t2; bne $t3, $zero, Overflow)

| instruction | Instr. address | Instruction format |
|---|---|---|
| strcpy: | | |
| (1) add $s0, $zero, $zero | 96400 | |
| (2) L1: add $t1, $s0, $a1 | 96404 | |
| (3) lbu $t2, 0($t1) | 96408 | |
| (4) add $t3, $s0, $a0 | 96412 | |
| (5) sb $t2, 0($t3) | 96416 | |
| (6) beq $t2, $zero, L2 | 96420 | ADR1 |
| (7) addi $s0, $s0, 1 | 96424 | 1 |
| (8) j L1 | 96428 | ADR2 |
| (9) L2: lw $s0, 0($sp) | 96432 | |
| (10) jal PRT_MSG | 96436 | ADR3 |
| (11) jr $ra | 96440 | 0 \| 31 \| 0 \| 8 |
| As procedure strcpy is invoked, initial registers are set as $a0 = 20000, $a1 = 24000, $sp = 60000 | | |

14. The left field of the above table lists a code sequence of 11 instructions for strcpy function. This code sequence is incomplete. Choose a right update from (a) or (b) to make this code sequence complete. The right field of the above table lists the MIPS instruction format. Based on the incomplete code sequence on the above table (not the updated code sequence), compute the values of ADR1 and ADR2. Which statements of (c) and (d) are correct?
    (a) Add the following instructions before instruction (1). (addi $sp ,$sp,-8; sw $s0,0($sp); sw $ra, 4($sp)); Add the following instructions before instruction (11). (lw $ra,4($sp); addi $sp, $sp, 8).
    (b) Add the following instructions before instruction (1). (addi $sp ,$sp,-4; sw $s0,0($sp);); Add the following instructions before instruction (11). ( addi $sp, $sp, 4)
    (c) ADR1 = 2;
    (d) ADR2 = 24101.




15. Consider the addition of two normalized binary floating-point numbers, $1.01111\times2^4$ (input 1) and $1.10011\times2^6$ (input 2). The significand bit of normalized binary floating-point number is hidden and exponent bias is not used. The accuracy of a binary floating-point number is to the fifth binary place, i.e., each fraction register is 5-bit long. The hardware design adopts the scheme of two additional bits, i.e., guard bit and round bit, to improve accuracy. Which of the following statements are true?
    (a) $c_0$= 1; $c_1$= 0; $c_2$= 1;
    (b) $c_3$= 2 (*imply* to perform a right shift by two bits); $c_4$= 0;
    (c) $c_5$= no operation; $c_6$= 0; $c_7$= no operation;

(d) $c_8$ triggers rounding hardware to produce a carry in of 1 to the fifth binary place; the final addition result is $1.11111\times2^6$.

16. A sequential CPU can be implemented in ***(i) single-(machine-)cycle-per-instruction***, ***(ii) multiple-cycles-per-instruction***, and ***(iii) pipelined*** fashions. For any particular machine program,
    (a) In ***(i)***, since every *instruction* must be completed in only one machine cycle, and since for any particular type of hardware circuits, if any instruction may possibly need to use it for any number of times, say, X, then X copies of it has to be in physical existence (meaning the hardware cost is at its maximum), hence the execution speed is the fastest among the three.
    (b) In ***(ii)***, all *instructions* are allowed to repetitively reuse any piece of hardware for any number of times; hence the required function units in this design can be the minimum.
    (c) In ***(iii)***, the CPU needs to execute many individual instructions simultaneously, with each instruction being in their respectively different pipe stages. Hence at the end of each pipe stage, a pipestage latch must be provided to keep the processing results of that pipestage. While pipestage latches in the ***(ii)*** design are not necessary (except for those storing the final outcomes of that instruction), since only one instruction will be executed in those cycles at any instance of time.
    (d) Both ***(ii)*** and ***(iii)*** design fashions have the same shortcoming of having to slow the machine clock down to allow for the slowest pipestage to be able to complete, making other faster stages to wait extensively without further productivity, hence are very rarely adopted in many successful commercial products.

17. We talk about the famous ***pipeline hazards*** in pipelined processor design, in their names' alphabetic order:
    (a) For ***Control hazards***, no matter how we restrict the branch condition complexity, how early we calculate the target (instruction) address and/or resolve the branch condition, how hard we predict the branch outcome and/or target address, pipeline stall(s) due to this kind of hazards are still unavoidable.
    (b) For ***Data hazards***, we mean the pipeline stalls or errors which may be caused by immature (不成熟的) read-after-write, arithmetic overflow, divide-by-zero, etc.
    (c) For ***Memory hazards***, we mean the pipeline stalls or errors which may be caused by data inconsistencies among different memory hierarchies, cache misses, page faults, errors occurring in data storage/transmission, etc.
    (d) For ***Structural hazards***, it can be dealt with by properly scheduling the uses of the contended (被爭奪的) hardware circuit and stalling the awaiting instructions, such that, for example, no matter how many instructions in the pipeline want to perform + operation at whenever for whatever number of times, only one hardware adder will suffice (滿足所需).

18. In a sequence of instructions in a user program, let instructions be numbered ..., ***i-1***, ***i***, ***i+1***, ..., and instruction ***i*** be later than ***i-1*** in execution order, etc. Assume that instruction ***i*** incurs (造成) an integer arithmetic overflow. Which of the followings can be a proper ratiocination (推理)?
    (a) The processor completes instructions up to ***i-1***, triggers exception handling to calculate the true result of instruction ***i*** and place the result properly; then the exception returns and resumes user program execution staring from instruction

*i+1*.

(b) The processor completes instructions up to ***i-1***, triggers exception handling to signal to the user of such an error; then the exception halts (停止) the user program.

(c) The processor ignores this overflow and simply continues the user program execution.

(d) Such an arithmetic overflow should not occur if this user program is generated by proper and commercially authorized (經授權或公認的) compiler and assembler, since such compilers and assemblers are designed to produce correct and high quality code.

19. Upon receiving a read/write request, a cache controller will partition the received memory address (assuming a byte-addressable address, since memories are byte-addressable by default) into three fields: ***tag***, ***index***, and ***byte-offset***.

(a) The size of a ***direct-mapped*** cache can be determined by knowing only the size of the ***tag*** field.

(b) The size of a ***set-associative*** cache cannot be determined based on the sizes of the three fields.

(c) For any reasonable cache, any of these three fields should not contain no bit (0 bit).

(d) If the same amount of memory is to be used for the ***fully-associative***, ***set-associative***, or ***direct-mapped*** caches, the ***tag*** field of the ***fully-associative*** cache is the largest, and that of the ***direct-mapped*** cache is the smallest.

20. The designs of ***cache memory*** and ***virtual memory*** both are based on a same principle. While the implementations of these memories are very different in some ways.

(a) The principle mentioned above is the ***Amdahl's Law***.

(b) The counterpart (對應的部分) of a ***cache block*** (or ***line***) in virtual memory design is called a ***page*** (or ***page frame***).

(c) During a ***cache miss*** or a ***page fault***, the CPU will typically stand still and wait until the miss or fault is handled.

(d) Just like in cache designs, both ***write-through*** and ***write-back*** polices are commonly used in a typical virtual memory system.

二、題組(20%)，共四個題組，每一題組內所有小題完全答對得 5 分，答錯任一小題或未作答得 0 分。題組 A 包含題號 21~24，題組 B 包含題號 25~27 ，題組 C 包含題號 28~30，題組 D 包含題號 31~33。

A. Consider the following set of processes performed on a single-core processor. These processes arrive at the following arrival time, with the length of the CPU burst given in the following table:

| Process | Arrival Time | CPU Burst Time |
|---|---|---|
| $P_1$ | 0 | 7 |
| $P_2$ | 2 | 4 |
| $P_3$ | 5 | 2 |
| $P_4$ | 7 | 6 |

Please calculate the average turnaround time U under the round robin scheduling with time quantum = 4. Assume $4*U= U_3*4^3 + U_2*4^2+U_1*4^1+U_0$ ($0 \leq U_0,U_1,U_2,U_3<4$).

Please choose the correct values of $U_3$, $U_2$, $U_1$, and $U_0$.

21. $U_3$=? (a) 3, (b) 2, (c) 1, (d) 0
22. $U_2$=? (a) 3, (b) 2, (c) 1, (d) 0
23. $U_1$=? (a) 3, (b) 2, (c) 1, (d) 0
24. $U_0$=? (a) 3, (b) 2, (c) 1, (d) 0

B. Consider the following system parameters of a microprocessor: A logical address is of 48 bits, and a physical address is of 32 bits. Let the page size be 16 KB. Answer the following questions:

25. What is the largest possible number of frames? (a^b stands for the b-th power of a. For example, 2^3=8).
    (a) 2^14
    (b) 2^18
    (c) 2^36
    (d) None of the above

26. Consider a simple, 1-level page table design. How many entries are there in a page table?
    (a) 2^14
    (b) 2^18
    (c) 2^36
    (d) None of the above

27. At least how many bits are necessary in a page-table entry?
    (a) 14
    (b) 18
    (c) 36
    (d) None of the above

C. The following two C and MIPS assembly programs perform the same task.

```
clear1(int array[], int size) {
  int i;
  for (i = 0; i < size; i += 1)
    array[i] = 0;
}
```

```
clear2(int *array, int size) {
  int *p;
  for (p = &array[0]; p < &array[size];
       p = p + 1)
    *p = 0;
}
```

```
       move $t0,ARY1
loop1: sll $t1,$t0,2
       add $t2,$a0,$t1

       sw $zero, 0($t2)
       addi $t0,$t0,ARY2
       slt $t3,$t0, ARY3

       bne $t3,$zero,loop1
```

```
       move $t0,PTR1
       sll $t1,$a1,2
       add $t2,$a0,$t1

loop2  sw $zero,0($t0)
       addi $t0,$t0, PTR2
       slt $t3,$t0, PTR3

       bne $t3,$zero,loop2
```

28. The ARY1 and PTR1 should be:
    (a) ARY1 = $zero; PTR1 = $a0
    (b) ARY1 = $a0; PTR1 = $zero
    (c) ARY1 = $a0; PTR1 = $a0
    (d) ARY1 = $zero; PTR1 = $zero

29. The ARY2 and PTR2 should be:
    (a) ARY2 = 1;  PTR2 = 1
    (b) ARY2 = 4;  PTR2 = 4
    (c) ARY2 = 1;  PTR2 = 4
    (d) ARY2 = 4;  PTR2 = 4

30. The ARY3 and PTR3 should be:
    (a) ARY3 = $t2;  PTR3 = $a1
    (b) ARY3 = $a1;  PTR3 = $a1
    (c) ARY3 = $t2;  PTR3 = $t2
    (d) ARY3 = $a1;  PTR3 = $t2

D. Consider cache placement and replacement policies. Assume that a code sequence of ***n+1*** instructions ***[i(0), i(1), i(2), ..., i(n-2), i(n-1), i(n)]***, where ***n*** is a power of 2, is to be executed repetitively for infinite number of times; Assume further that all ***instructions*** are 32 bits long, and the ***cache lines*** (or ***blocks***) are also 32 bits long. (Hint: To help derive the answers in following questions, you may let n=4, a small number.)

31. If a ***direct-mapped*** cache of ***size*** n-cache lines is used, the resulted cache ***hit rate*** is about:
    (a) (n-2)/n
    (b) (n-1)/n
    (c) (n-2)/(n+1)
    (d) (n-1)/(n+1)

32. If a ***fully associative*** cache of the same ***size***, with the commonly used ***LRU*** (***least-recently-used***) ***replacement***, is used, the resulted cache ***hit rate*** is about:
    (a) 0
    (b) (n-1)/n
    (c) (n-2)/(n+1)
    (d) (n-1)/(n+1)

33. If a ***fully associative*** cache of the same ***size***, with a less famous ***MRU*** (***most-recently-used***) ***replacement***, is used, the resulted cache ***hit rate*** is about:
    (a) 0
    (b) (n-1)/n
    (c) (n-1)/(n+1)
    (d) n/(n+1)