答題注意事項/Instructions to answer the questions :

1) 考題分為【單選題】與【複選題】。各題的配分不完全相同，分別標示。
   There are two types of questions: **Multiple Choices** and **Multiple Choices with at least one correction choice**.
   Each question has its own individual score.

2) 【單選題】：只有一個選項為正確答案。
   Multiple Choices: only one among the four or five choices is the correct answer.

3) 【複選題】：至少有一個選項為正確答案。每一個選項分別計分。錯誤選項為零分，不倒扣。
   Multiple Choices with one or more correction answers: at least one among the four or five choices is the correct answer.
   Each choice will be graded individually. No penalty for incorrect selection.

4) 本考試科目包含兩個子科目：作業系統與計算機結構。配分各為五十分，總分一百分。
   This subject consists of TWO sub-subjects: Operating Systems and Computer Architecture. Each sub-subject has fifty points and the total points are one hundred.

5) 各子科目均包含【單選題】與【複選題】，分別標示。
   Each sub-subject has Multiple Choices and Multiple Choices with at least one correct choice. They are labeled in each sub-subject.

6) 本題目卷總計有 11 頁。
   The question sheets have 11 pages.

※ 注意：請用 2B 鉛筆作答於答案卡，並先詳閱答案卡上之「畫記說明」。

見背面

作業系統 (50 分) /Operating Systems (50 points)

【單選題】只有一個選項為正確答案。
Multiple Choices: only one among the four or five choices is the correct answer.

1. (5 points) Which **ONE** of the following statements is correct for the critical-section problem? Assume there are several concurrent processes, each of which has one critical section. Memory variables shared among these processes are protected by the critical section from concurrent updates.
   a. Memory barriers can prevent concurrent updates by enforcing mutual exclusion to the shared memory.
   b. Atomic instructions can prevent concurrent updates by completing the load and store instructions to the shared memory.
   c. Semaphores can allow more than one process to update the shared memory.
   d. Mutex locks acquire the lock before updating the shared memory and release the lock after updating the shared memory when the locks are mandatory to assure mutual exclusion.
   e. None of all.

2. (5 points) Which **ONE** of the following statements is correct for the redundant array of inexpensive disks (RAID)?
   a. Reliability of the storage system is increased by the parallelism of the multiple disks.
   b. RAID 5 uses the block-interleaved distributed parity check to increase the reliability.
   c. If the MTBF of a single drive is 100,000 hours and the mean time to repair is 10 hours, the mean time to data loss of a mirrored drive (two disks) system is 5,700 years.
   d. RAID 10 refers to a combination of RAID levels that are stripped first and mirrored later.
   e. None of the above.

3. (3 Points) Which **ONE** of the following statements is correct? Please select the correct description for the output of the following program.

```c
#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
int value = 10;
int main()
{
    pid_t pid;
    pid = fork();

    if (pid == 0) {
        value += 10;
        printf("A = %d\n",value);
        return 0;
    }
    else if (pid > 0) {
        wait(NULL);
        print("B = %d\n",value);
        return 0;
    }

}
```

接次頁

Choices:
(A) $A+B^2=120$
(B) $A=B$
(C) $2A+3B=50$
(D) $B=20$
(E) None of the above.

4. (7 Points) Given the following descriptions, which **ONE** of the following statements is correct? Please calculate the values of variables **P, L, X, N, Y, A, and B** to select the correct statement.

Consider a paging system with a page size of 2KB. The amount of physical memory is 64KB. Let **P** denote the number of bits required in the physical address. Let **L** denote the number of bits required in the logical address.

Conventionally, a single-level page table is stored in the memory. If the page table is implemented in the conventional way, a page table in the memory contains 1024 entries. The memory access time is 180 nanoseconds (ns). Assume that the hit ratio of the conventional page table is 60% and the average page fault service time is 30,000 microseconds (μs). Therefore, the effective memory access time for the conventional approach with a single-level page table is **X**.

Alex would like to consider an *"inverted page table"* to improve the translation from a logical address to a physical address. After careful calculation, he found that the inverted page table requires **N** entries. Furthermore, Alex considers the translation look-aside buffer (TLB) hardware to implement the idea of the inverted page table. Searching TLB takes 20 nanoseconds (ns). Assume that the hit ratio of the inverted page table is 90%. The effective memory access time for Alex's approach with a TLB-implemented inverted page table is **Y**. So, Alex's approach improves the effective memory access time, where **A< speedup** (in terms of the effective memory access time**) <B.** Here, A and B are positive integers.

Choices:
(A) $2L+3P+5N+A^3+B^2=293$.
(B) $AB(5L+2P+N)>10000$
(C) $4L-2P+3N-5A+2B=205$
(D) $3L+PN-4A+2B>1024$
(E) None of the Above.

5. (10 Points) Given the following descriptions, which **ONE** of the following statements is correct? Please calculate the values of **P, D, M, and F** to select the correct statement.

Assume that we have a computer system with a 32-bit processor. The computer supports a paged virtual memory. The page size is 8KB. Each address referenced by the processor contains **P** bits for indexing the page number and **D** bits for indexing the page offset. Initially, three empty frames are given to execute a process. Assume that the processor of the computer references the following sequence of hexadecimal addresses to execute the process:

4DF6BC24
1CA54523
4DF6AC24
1CA55523
23A22260

見背面

23BCDFFF
23A23261
23BCCFFF
1CA54523
23B23261

The working-set model is implemented in the system. The size of the working-set window is W=6. Generally, processes tend to have locality of memory references. Alex looked over the sizes of working sets resulting from the above sequence of referenced addresses. He found that the maximal size of working sets over the referenced sequence is **M**. Therefore, Alex designed a locality-based page replacement algorithm according to the locality principle. To implement his algorithm, he quantifies "*referenced likelihood for each page*" as the ratio of the total number of references to the page within the working-set window to the size of the working-set window. Note that the pages referenced within the working-set window include the page that the processor is currently attempting to reference. When a hexadecimal address is referenced, the algorithm needs to track the referenced likelihood for each page. If a page fault happens, the page with the minimal referenced likelihood is selected to be replaced. If multiple pages have the same referenced likelihood, the Least recently used (LRU) algorithm is used to break the tie. So, the total number of page faults arising from the sequence of hexadecimal addresses referenced by the processor is **F**.

Choices:
(A) $2P+5D+M^4+F^2>1000$
(B) $M+D+P+F=42$
(C) $5M+2D+2P-F=34$
(D) $M^2-D^2+3P+4F>500$
(E) None of the above.

【複選題】： 至少有一個選項為正確答案。每一個選項分別計分。錯誤選項為零分，不倒扣。

Multiple Choices with one or more correction answers: at least one among the four or five choices is the correct answer. Each choice will be graded individually. No penalty for incorrect selection.

6. (5 points) The following figure shows an example data structure used for opening a file from file systems in user space, kernel memory, and secondary storage in operating systems. The number in the circle represents one operation in the computing systems.

| user space | Kernel Memory | secondary storage |
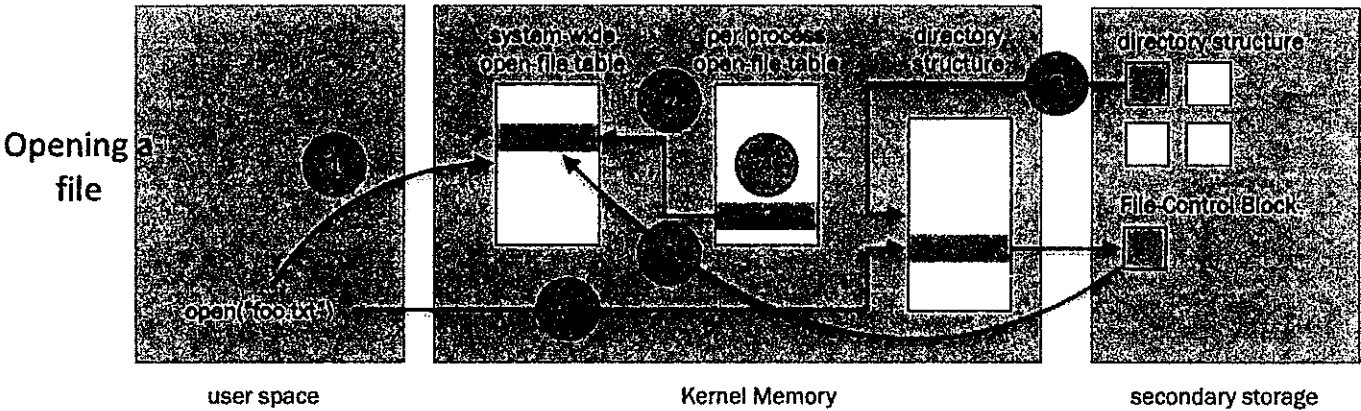
Select the correct statement(s) for the above figure.
(A) Opertion ① will be conducted to ensure that there is no duplicated file in the same directory.
(B) Opertion ② returns the file descriptor to the system-wide open-file table to create an entry.
(C) When the file "foo.txt" does not exist in the file system, Opertion ③ is called to create a new entry in the directory structure.
(D) When the file 'foo.txt' has been open in the system, Opertion ⑤ will not be called.
(E) When the file "foo.txt' has not been open in the file system, Opertion ④ creates an entry to store the File Control Block of the file.

7. (5 points) Given a set of 5 processes, the arrival time of the i-th process P_i is A[i], and the burst time of the i-th process P_i is B[i], as shown below:

Process	Arrival Time, A[i]	Burst Time, B[i]
P_1	0	10
P_2	0	8
P_3	0	5
P_4	0	2
P_5	0	4

Perform **round-robin** scheduling for these five processes. Assume that the average waiting time of all processes is a $\times 10^2$ + b $\times 10^1$ + c $\times 10^0$ + d $\times 10^{-1}$. Select the correct description(s).

Choices:
(A) a + b + c + d = 8.
(B) b = 2c.
(C) b + d = 3.
(D) 5b = c.
(E) a x c = 5.

見背面

8. (5 points) Given a set of 5 **periodic** processes, which all arrive at time 0, as shown below:

Process	Relative Deadline	Period	Burst Time
P_1	10	50	10
P_2	10	120	10
P_3	5	800	5
P_4	10	20	10
P_5	10	40	10

Perform the **deadline-monotonic scheduling**. Select the correct description(s).

(A) Deadline monotonic algorithm and rate monotonic algorithm have the same schedulability for this task set.

(B) The worst-case response time (from the arrival time to the time of the completion of a process) of P_2 is 20.

(C) Process P_4 is schedulable.

(D) If we increase the burst time of P_2 by X time units, the worst-case response time of P_2 will also be increased by X time units.

(E) All the tasks remain schedulable when the burst time of process P_4 is set to 8.

9. (5 Points) Please select the correct description(s).

(A) A system with two processing cores can always get a double speedup.

(B) For process execution, it is possible to have concurrency without parallelism.

(C) The Pthreads API(s) can be used in multithreaded C programs for thread creation and synchronization. When the parent thread calls the pthread_join() function, it waits for the child thread to terminate. It is synchronous threading.

(D) A process transits from a waiting state to a running state when it is assigned CPU for execution.

(E) If the binding of instructions and data to memory addresses (i.e., address binding) is done at the compile time, it is called "dynamic binding".

接 次 頁

計算機結構 (50 分) /Computer Architecture (50 points)

【複選題】: 至少有一個選項為正確答案。每一個選項分別計分。錯誤選項為零分, 不倒扣。

Multiple Choices with one or more correction answers: at least one among the four or five choices is the correct answer. Each choice will be graded individually. No penalty for incorrect selection.

10. (8 points) Given the following C code snippet and the disassembly of the resulting object of the code for the x86-64 platforms. Assume for *func(i, up): i is in %rdi, up is in %rsi*. Answer the value of *CN* and the full declaration of the structure *np_struct*.

HINT: for an instruction *lea (a, b, num), c*, the result of *(b*num + a)* is moved to *c*.

```
                                    extern struct np_struct;
mov    0x120 (%rsi), %ecx           typedef struct {
add    (%rsi), %ecx                    int fst;
lea    (%rdi, %rdi, 4), %rax           nt_struct a[CN];
lea    (%rsi, %rax, 8), %rax           int lst;
mov    0x8 (%rax), %rdx              } u_struct;
movslq %ecx, %rcx
mov    %rcx, 0x10 (%rax, %rdx, 8)
retq
                                    void func(long i, u_struct *up) {
                                        int n = up->fst + up->lst;
                                        np_struct *np = &up->a[i];
                                        np->x[np->idx] = n;
                                    }
```

Choices:
(A) CN = 8, nt_struct = {long x[3], long idx};
(B) CN = 7, nt_struct = {long idx, long idx1, int x[4]};
(C) CN = 8, nt_struct = {long idx, int x[5]};
(D) CN = 7, nt_struct = {long idx, long idx1, long x[3]};

11. (5 points) Two different compilers were used to compile the same program. When using compiler A, the program executed 1.0E9 instructions and took 1.5 seconds to complete. With compiler B, the program executed 1.5E9 instructions and took 1.8 seconds to complete. Assume the processor has a clock cycle time of 1ns.

(A) The average CPI of the program produced by compiler A is 1.5.
(B) The average CPI of the program produced by compiler B is 1.5.
(C) Assume that the compiled programs now run on two new distinct processors. If the execution times on the two processors are the same, the clock of the processor running compiler A's code is 1.37x faster than the clock of the processor running compiler B's code.
(D) Assume that a new compiler, C, is developed using 4.0E8 instructions and achieves an average CPI of 0.8. The speedup of using the new compiler versus compiler A is less than 4.7x.

見背面

(E) The speedup of using compiler C described in (D) versus compiler B is less than 4x.

12. (6 points) Assume the individual stages of the datapath have the following latencies:

   IF: 150ps; ID: 250ps; EX: 50ps; MEM: 200ps; WB: 100ps

   Assume the instructions executed by the processor break down as follows:
   ALU: 45%; BEQ: 20%; LW: 20%; SW: 15%

   Choices:
   (A) The clock cycle time in a pipelined processor is 350 ps.
   (B) The total latency of an LW instruction in a non-pipelined processor is 850 ps.
   (C) The total latency of an LW instruction in a pipelined processor is 1250 ps.
   (D) The utilization of the data memory is 35% (Assume there are no stalls or hazards).

【單選題】只有一個選項為正確答案。
Multiple Choices: only one among the four or five choices is the correct answer.

13. (5 points) Consider a processor with a load delay slot: this means that the value loaded into a destination register (Rd) is not available for use in the instruction immediately following the load. Assume the following code sequence runs on this processor (with RISC-V ISA).

```
addi x3, x0, 0x18 //x0 is the zero register
lw x3, 8(x3)
add x4, x3, x3
```

Suppose that MEM[32] = -4, and the load delay slot is *ignored*.　What will be the value (in decimal) stored in the register x4 after all the code is executed?

    (A) 48
    (B) 14
    (C) -8
    (D) 40

(Question Group A: 8points – 2 questions) Assume we have a 2-way set associative cache that employs a write-back policy and uses a perfect LRU mechanism to decide which block to evict. The tag store, which includes bits for valid status, dirty status, and LRU tracking, requires 13×2^8 bits of storage space. The cache implements virtual indexing with physical tagging. The system has a virtual address space of 1 MB, uses pages of 1 KB in size, and each cache block is 8 bytes long. Answer the following questions.

14. (4 points) What is the size of the cache's data store?
    (A) 1KB
    (B) 2KB
    (C) 4KB
    (D) 8KB

15. (4 points) How many bits of the virtual index come from the virtual page number?
    (A) 1 bit
    (B) 2 bits
    (C) 3 bits
    (D) 4 bits

(Question Group B: 12points – 2 questions) Assume the following C code is given:

```
for (i=2; i<=1100; i++)
    A[i] = A[i-1] + A[i-2];
```

The RISC-V assembly respective to the C code is:

```
li x5, 8800
```

int : 8 bytes

見背面

```
                (100)
        add x12, x10, x5
        addi x11, x10, 16
LOOP:           λ   A =2
        fld f0, -16(x11)  -2
        fld f1, -8(x11)  -1
        fadd.d f2, f0, f1
        fsd f2, 0(x11)  load to A[i]
        addi x11, x11, 8  λ+1
        ble x11, x12, LOOP
```

Here, an instruction's latency represents how many clock cycles must pass before another instruction can use its result. The floating point instructions have the following associated latencies (in cycles):

fadd.d = 4
fld = 6
fsd = 1

$$(100-2+1) * (6+6+4+1)$$
$$= 1099 * 17 =$$

16. (4 points) How many clock cycles are needed to execute this code?
    (A) 15984
    (B) 17587
    (C) 25280
    (D) 25283

$$\underset{+9}{} $$
$$1100 * 17 - 17$$
$$= 18706 - 17 = 18683$$

17. (8 points) Storing and re-loading data from the main memory during the loop cause a slowdown.

Assume now that you have a new instruction called *fmv.d* that executes in a single cycle. This instruction performs register move operations. For instance: "*fmv.d rd, rs1*" — writes the value of floating-point register *rs1* into the floating-point register *rd*.

If the *fmv.d* instruction is used to optimize the code's performance by using registers to carry the data between iterations of the loop, how many cycles are needed to execute this code?
Hint: think of what the assembly will look like after applying *fmv.d* and count the total cycles the resulting code takes.
    (A) 7703
    (B) 7806
    (C) 7823
    (D) 6934

18. (6 points) Consider a computer system equipped with a single CPU with the following organization: It contains two cores, where each core implements superscalar processing with two functional units and can execute instructions out of order. The cores operate in single-threaded mode, meaning each core can only execute one thread at a time.

Two threads, X and Y, must be executed on this CPU. Each thread consists of the following sequence of operations: Unless noted or encountered a hazard, you can assume all operations take a single cycle to execute.

接 次 頁

X	Y
A1: takes two cycles to execute	B1: no dependencies
A2: depends on the result of A1	B2: conflicts for a functional unit with B1
A3: conflicts for a functional unit with A1	B3: no dependencies
A4: depends on the result of A2	B4: depends on the result of B2

How many cycles will it take to execute these two threads? How many issue slots are wasted because of hazards?

(A) 4 cycles, 7 slots wasted
(B) 4 cycles, 3 slots wasted
(C) 5 cycles, 5 slots wasted
(D) 6 cycles, 7 slots wasted

(以下空白)

試題隨卷繳回