NOTE that in the question, it is intended to provide redundant or miss certain assumption to disguise you. Please make your own assumption if necessary to answer the questions.

1. True or False? Answer TRUE or FALSE for the following statements.
   a) [2 pt.] You must have multi-issue machines to benefit from software-pipelining or global instruction scheduling.
   b) [2 pt.] Two instructions with data dependencies will cause pipeline stall(s) in execution.

2. Multiple Choices (One correct answer only)
   a) [4 pt.] Increasing the page size from 4 KB to 2048 KB on modern Intel/AMD x86-64 CPUs has reduced one application's execution time by 40%. The measured working set size of this application is about 2 GBytes. Assume the swap is disabled and this application randomly accesses 4 bytes at a time over the entire working set repeatedly. Select one main reason from below.
      1) Prevents the need for pipeline bubbles.
      2) Improves the efficiency of adjacent cache line prefetcher.
      3) Keeps data in a more continuous address in physical memory to reduce memory access latency.
      4) Reduces the number of completed page walks.
      5) Reduces the rate of page fault.

   b) [4 pt.] Which one is FALSE for an out-of-order execution superscalar CPU?
      1) It is typically also pipelined.
      2) It may execute multiple instructions per clock cycle.
      3) It will check data dependencies between instructions dynamically at run time.
      4) It exploits data-level parallelism primarily by the out-of-order execution mechanism.
      5) It could be a RISC or CISC CPU.

   c) [4 pt.] Security architecture is important. Many smartphones support TEE (Trusted Execution Environment). What function below does NOT make sense to be put in the TEE secure area of your smartphone?
      1) Fingerprint authentication, which needs to access the fingerprint sensor hardware
      2) Blockchain cryptocurrency mining, as in the recent HTC's blockchain phone
      3) Storing my bitcoin wallet's private key
      4) Supporting Android keystore system
      5) Random number generation

3. ISA (Instruction Set Architecture)

見背面

Pre-increment and post-increment addressing modes are often adopted in processor ISA such as x86 and Itanium. In ARM, the post-increment load instruction is like

```
ldr r1, [r2], #4    // r1<--mem[r2], r2<--r2+4
```

However, this is actually a pseudo instruction in ARM. In x86, push and pop are real pre-increment and post-increment instructions. The pre-increment instruction changes the base address in a register by the offset value, and then performs the data transfer using the new address in the register. The post-increment instruction performs the data transfer first, then changes the base address register.

a) [2 pt.] What is a pseudo instruction? What is the advantage of having pseudo instructions?
b) [3 pt.] What are the advantages to have pre-increment and post-increment real instructions? What type of high level language code structures would benefit from them?
c) [1 pt.] Why Intel/HP Itanium supports only post-increment instructions but not pre-increment instructions?
d) [2 pt.] What are the disadvantages of having such instructions in the ISA?

4. Exceptions and Interrupts.

a) [2 pt.] Which of the follow are considered as exceptions and which are considered as interrupts.?

Page fault
TLB miss
Floating point arithmetic underflow
I/O device request
Undefined instruction
User defined interrupt
Execution abort
System call

b) [3 pt.] Some microarchitectures (implementation) are more difficult to handle exceptions, please rank the following implementations in terms of difficulties in handling exceptions, rank from the most difficult one to the simplest one.

Pipelined implementation
Superscalar implementation
Out-of-order superscalar
Speculative execution
Hierarchical data caches
Single issue In-order processor

接次頁

c) [3 pt.] Which of the following are most difficult to handle interrupts., please rank from the most difficult one to the simplest one.

GPGPU
Hyper-threaded processor
Pipelined processor
Superscalar processor
Containers
Virtual Machines

5. Parallel Execution

a) [2 pt.] What are the meanings of the following acronyms?

ILP
DLP
MLP
TLP

b) [4 pt.] For the above four items in (a), give at least one architecture/ microarchitecture technique that can effectively exploit each of them.
c) [3 pt.] For the above four items in (a), give at least one software technique that can increase the degree of parallelism for each of them.

6. Multi-core

[9 pt.] Many of your smart phones use 8-core Qualcomm processors. For example, the recent Snapdragon 670 and 850 processors both have 8 cores. Let us say the 8-core processor has B big cores and L little cores, where B + L = 8. For Snapdragon 670, $B_{670} = 2$ and $L_{670} = 6$. For Snapdragon 850, $B_{850} = 4$ and $L_{850} = 4$. Big core: consumes more power and deliver higher performance than little cores. Assume you are in charge of the design of an 8-core Processor 2020 of Year 2020 and

a) If I suggest to you that $B_{2020} = 6$ and $L_{2020} = 2$, what will be your response? Agree to or object to it? Explain the rationale behind your response. You get 0 point if you don't explain.
b) In Processor 2020, you will have a CPU_CLUSTERbig and a CPU_CLUSTERlittle; Explain why and how to maintain cache coherence between the two clusters.

見背面

7. (25 pt.) The logic and physical computation resources in computing systems are shared among processes and threads.

a) (10 pt.) In order to control the access rights of the processes in a computer, process IDs are assigned to every process. Please answer the following questions.
   1) (6 pt.) Please describe the property of the following three IDs of one process and their difference: **real UID**, **effective UID**, and **saved UID**.
   2) (4 pt.) When one process requires additional permission to access certain resources, which of the three aforementioned IDs should be changed?

b) (15 pt.) Multi-tasking is well supported in modern operating systems. Please answer the following questions.
   1) (5 pt.) Please describe the difference of using multi-process and multi-thread for multi-tasks from the perspective of memory usages, CPU scheduling, and resource sharing.
   2) (3 pt.) Please read the following code segment in C for multi-processing. Will the variable printed on Line 24 be 'C'? If yes, please describe the reason of changing from 'P' to 'C'. Otherwise, please describe the cause of having different values in two processes.

```
 1 #include <stdio.h>
 2 #include <sys/types.h>
 3
 4 int main(void)
 5 {
 6    pid_t pid;
 7    char sharedVariable='P';
 8    char *ptrSharedVariable;
 9    ptrSharedVariable=&sharedVariable;
10    printf(" Address  is %p\n",ptrSharedVariable);
11    printf(" char value is %c\n",sharedVariable);
12    pid = fork();
13    if (pid == 0)   {
14       sharedVariable='C';
15       printf(" *** Child process  ***\n");
16       printf(" Address  is %p\n"  , ptrSharedVariable);
17       printf(" char value is %c\n",sharedVariable);
18       sleep(5);
19    }
20    else {
21       sleep(5);
22       printf("\n *** Parent process ***\n"  );
23       printf(" Address  is %p\n"  , ptrSharedVariable);
24       printf(" char value is %c\n"  , sharedVariable);
25    }
26 }
```

   3) (3 pt.) Are the addresses printed on Line 16 and 23 same from each other? If yes, please describe the reasons of having the same value. Otherwise, please describe the cause in details.

接次頁

4) (4 pt.) What's the results of Line 24 after fork() on Line 12 is replaced by vfork()?

8. (25 pt.) On Singles Day (光棍節), millions of on-line shopping transactions are processed in a very short period of time. The supply of each product is limited. We need to make sure there is enough supply for each commit transaction and processing should be as soon as possible.

a) (5 pt.) What kind of transaction scheduling algorithm can generate the most revenue for the shopping service provider (such as PCHome)?
b) (5 pt.) How do we guarantee that the earlier transaction is the earlier processed? Hint: The earlier sent or arrived? There is no global clock.
c) (5 pt.) If we would concurrently process the transactions in a distributed manner, how to synchronize so that the products are not over sold while the process time is minimized as possible.
d) (5 pt.) The transactions might be cancelled by users for any reason or by the system because there is no enough supply. How do we design the file system to store transactions so that it is both trustful and efficient?
e) (5 pt.) What attacks need to be handled at this high-speed transaction? How?

試題隨卷繳回