1 選擇題 (2 pts each):

(1) After the interrupt handler processes an interrupt, what happens?

A) The OS checks between each instruction to see if an interrupt occurred.

B) The OS checks to see which controller caused the interrupt.

C) The interrupt handler returns to the interrupted program in user mode.

D) The hardware jumps to the handler through the interrupt vector.

E) None of the above happens after a controller generates an interrupt.

(2) In OSs for modern machines, what is the immediate result if a process generates an address that is outside its legal address range?

A) Blue Screen of Death

B) A hardware interrupt (trap) will occur

C) The program will probably not work right

D) The O/S call will return an error

E) None of the above is the immediate result of generating an illegal address.

(3) What is the biggest problem with the FCFS CPU scheduling algorithm?

A) It is hard to implement

B) It is not fair

C) A long job can delay shorter ones

D) It is not intuitive to understand

E) None of the above is true about the FCFS algorithm.

(4) What is the problem with the Shortest Run Time First process scheduling algorithm?

A) We don't know the next CPU burst length.

B) It may lead to starvation.

C) It can make shorter jobs wait behind longer jobs.

D) It uses too much CPU time to run.

E) None of the above is a problem with SRTF job scheduling.

(5) What does the OS Call to Fork a Process ask the OS to do?

A) start a new copy of the running process

B) start a program running as another process

C) kill a running process

D) create a new empty process but do not start any program running in it yet

E) None of the above is what a fork call does.

(6) What system call does a Linux job normally use to start another copy of itself?

A) `fork/exec`

B) `shell`

C) `clone`

D) `dispatch`

E) None of the above is used by a Linux job use to start another copy of itself.

(7) We said that we could prevent some deadlocks by using which of these techniques?

A) Periodically check for a loop in the wait states.

B) Always check for a safe state before granting resources.

C) Preempt resources from processes.

D) Agree about the order to apply locks in.

E) None of the above will prevent any deadlocks.

(8) Shared memory is a very fast mechanism for interprocess communication because both processes can see any changes instantly. What is the main drawback to shared memory?

A) It is limited to a small space.

B) It requires an OS call.

C) It is limited to two processes at one time.

D) It may require synchronization.

E) None of the above is a drawback to shared memory.

(9) With paged memory hardware there was a bit in each page table entry called the “valid” bit that originally meant the page was not a part of the logical address space for the process. When Virtual Memory was implemented on top of this hardware we changed the significance of that bit. What did it now mean?

A) The page was not currently in memory.

B) The page had not been reference lately.

C) The page was read only.

D) The page was part of the kernel space.

E) None of the above describes the meaning of the “valid” bit under VM.

(10) When data is transferred between memory and an I/O device the transfer is often done in blocks. There are several reasons why we might do block transfers. Which of the following is not a reason for using block transfers?
A) Only one interrupt per block.
B) Exploit the parallelism of the bus
C) Spread the overhead of accessing the media (e.g., tape start-stop time)
D) Exploit the principle of locality of reference
E) All of the above are reasons why we might do block transfers.

2 (5 pts) Consider a file system that uses inodes to represent files. Disk blocks are 8-KB in size and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, plus single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system?

3 (10 pts; 5 pts each) Rate Monotonic Scheduling (RMS) is a priority assignment algorithm used in real-time operating systems with a static-priority scheduling class. The priority of a process is assigned based on the inverse of its period. On the other hand, Earliest Deadline First Scheduling (EDF) is also a priority assignment algorithm in real-time operating systems with a dynamic-priority scheduling class. Priorities are assigned according to deadlines: the earlier the deadline, the higher the priority; the later the deadline, the lower the priority. In the hard real-time environment, any task must be serviced by its deadline. That is, every task should acquire enough processing time to finish its execution before its deadline/period, which is also referred as "meet the deadline requirement". Suppose there are four real-time processes in the system, as shown in the table. Please answer the following questions:

Process	Processing Time	Period
P1	10	50
P2	20	100
P3	50	150
P4	80	300

(a) Suppose the Rate Monotonic Scheduling algorithm (RMS) is adopted to schedule the real-time process set, and all the processes are ready at time 0. Assume the deadline for each process equals to its period. Can all the process meet their deadline requirements?

(b) Suppose that the Earliest Deadline First Scheduling (EDF) is adopted to schedule the real-time process set, and all the processes are ready at time 0.

Assume the deadline for each process equals to its period. Can all the process meet their deadline requirements?

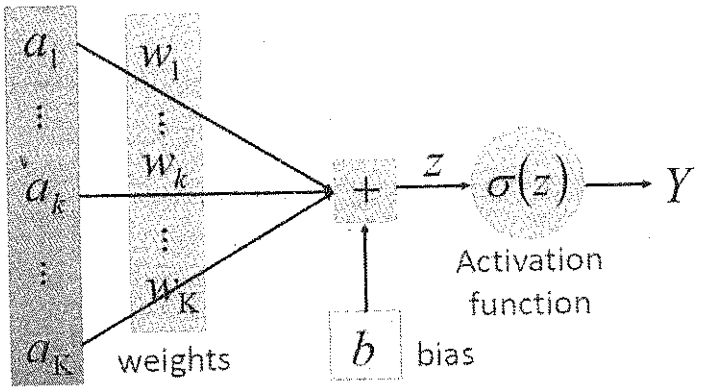
4 (15 pts; 5 pts each) Given a computer system with 64-bits virtual address and 48-bits physical address, and the system is word-addressable (Suppose 1 word = 2 bytes). Suppose each entry in the page table takes 8 bytes to record the management information, and the page size is 8KB.

(a) What would be the maximum number of pages owned by a process?

(b) Suppose that we have multi-level paging. How many levels do we have in multi-level paging?

(c) Assume that the Inverted Page Tables is used to reduce the DRAM consumption. Also, assume that each entry in the inverted page table takes 8 bytes to record the mapping information. What is the amount of DRAM (bytes) required for the whole inverted page table?

5 (10 pts) In the following simple neuron network, there are **K** inputs (a_1 to a_K), **(K+1)** model parameters (w_1 to w_K and b) and one neuron network output (Y). σ(z) is a non-linear sigmoid function which modeled by a second-order polynomial function. The model parameters (w_1 to w_K) are loaded before computing the neuron network output. If we assume the delay times for a two-input adder and a two-input multiplier are **DADD** and **DMUL**, respectively. Also, **DADD** is equal to 0.1× **DMUL**. If there are 16 inputs (**K=16**), Please determine the minimum delay time for this neuron network in terms of **DMUL**. (Hint: You can perform multiplications and additions in parallel to minimize the delay time)



$$z = a_1 w_1 + a_2 w_2 + \cdots + a_k w_k + \cdots + a_K w_K + b$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \approx D + Ez + Fz^2, \quad Y = \sigma(z)$$

6 (10 pts) For the same neuron network shown in problem 1, if a CPU is used to compute the neuron network output. Addition, the hardware resource limitation restricts only one multiplication and one addition can execute in parallel in one clock cycle. Please determine the minimum clock cycles to compute the output of this neuron network when inputs are changed.

7 (5 pts) Hit time, miss rate and miss penalty are three metrics for cache optimizations. For each metric, please give one method for cache optimization.

8 (25 pts; 5 pts each) The main memory is byte-addressable and the CPU is going to access the following 12 addresses: 8, 20, 182, 88, 39, 40, 98, 182, 57, 32, 66, 88 (in decimal).

(a) Assume CPU uses these addresses to read 12 32-bit variables. Which is the first non-aligned memory access?

(b) Assume CPU uses these addresses to read 12 8-bit (1-byte) variables, and a direct-mapped cache exists between CPU and main memory. If the cache has 10 blocks, each of which can only hold 1-byte data, which is the first read access that has a "cache hit"?

(c) What is the total number of cache hits for the 12 read accesses in (b)?

(d) If each cache block in (b) can hold 10-byte data (i.e. the block size now becomes 10-byte), which is the first read access that has a "cache hit"?

(e) What is the total number of cache hits for the 12 read accesses in (d)?