Note: Keep your answers brief and precise.

Answer all questions.

Group A: Short Answer Questions [22 Marks]

1. (22 points) True/False. Justify your answer. Note: No credit will be given for unjustified answer.
   (a) The OS scheduler is invoked when a process finishes execution.

   Solution: true. The kernel will gain control of the CPU so it can select another process to execute on it.

   (b) Two devices can each generate interrupts at the same time.

   Solution: true. This is why an interrupt manager is needed.

   (c) The kernel gets back CPU control, when a process makes a blocking read() call.

   Solution: true. The kernel will gain control of the CPU so it can select another process to execute on it.

   (d) A semaphore is a synchronization primitive that can be used in user level threads.

   Solution: false. Semaphores requires system calls and therefore it is not suitable for thread libraries used by ULT.

   (e) In programs, we can use pthread locks in place of semaphores to perform the same functionality.

   Solution: false. Semaphores can be used for synchronization in addition to mutual exclusion. However, locks can only be used for mutual exclusion.

   (f) Suppose that a multi-threaded program requires a lot of I/O, it is better to (from program execution time) to have kernel-level threads.

   Solution: True. Kernel-level is better so that if a thread gets blocked by an I/O request (which is usually a system call), other threads can continue executing.

   (g) A CPU bound process that is allocated memory pages fewer than its working set will behave like an I/O bound process.

   Solution: true. if a process is allocated memory pages that are a lot fewer than its working set, then thrashing will occur and many page faults will happen. The process will be blocked so often and will behave like an I/O bound process.

(h) The purpose of the TLB is to place the contents of the most frequently accessed page frames in the L2 cache of the CPU.

Solution: false. The TLB is a dedicated cache.

(i) In a computer system with a single CPU and we do not know the runtime of each process, we can use the Shortest Remaining Time Next scheduling.

Solution: False. The SRTN algorithm requires knowledge of the estimated execution time of each job.

(j) Contiguous allocation of files leads to external disk fragmentation.

Solution: True. Contiguous allocation leads to external disk fragmentation since disk are divided into blocks the only disk space waste is in the block itself.

(k) The number of disk operations needed to fetch the i-node for a file with the path name `/home/cs333/final.txt` is 10. Assume that the i-node for the root directory is in memory, but nothing else along the path is in memory. Also assume that all directories fit in one disk block.

Solution: False. The correct number is 6 . The procedure is as follows:

- read the root directory of `/` and scan it for the directory user (this will identify the location of the i-node of `usr`
- read the i-node for `home` and identify the address of its block on disk
- read the directory of `home`
- read the i-node for `cs33`
- read the directory of `cs33`
- read the i-node for `final.txt`

Group B: Processes, Threads, and Synchronization[16 Marks]

2. (4 points) Processes can be in one of three states: Running, Ready, or Blocked. In which state is the process for each of the following four cases?
   (a) Waiting for data to be read from a disk.

   Solution: Blocked

   (b) Busy waiting for a lock to be released.

   Solution: Running

   (c) Having just called wait() on a condition variable and the condition is not satisfied.

   Solution: Blocked

   (d) Having just completed an I/O and waiting to get scheduled again on the CPU.

   Solution: Ready

3. (4 points) Suppose that two processes: P1 and P2 are running for a long time in a system. Neither process performs any system calls that might cause it to block, and there are no other processes in the system. P1 has 2 threads and P2 has 1 thread.
   What percentage of the CPU time will P1 get in the following two cases:
   (a) Threads of the processes are user level threads (ULT).

   Solution: If the threads are user threads, the threads of each process map to one kernel thread, so each process will get a share of the CPU. The kernel is unaware that P1 has two threads. Thus, P1 will get 50% of the CPU.

   (b) Threads of the processes are kernel level threads (KLT).

   Solution: If the threads are kernel threads, they are independently scheduled and each of the three threads will get a share of the CPU. Thus, the 2 threads of P1 will get 2=3 of the CPU time. That is, P1 will get 66% of the CPU.

4. (3 points) List three causes for switching from a user process into the OS kernel.

Solution: 1- System calls, 2- Exceptions, 3- Interrupts

5. (2 points) How does a system call differ from a normal function call? Identify two distinct differences.

Solution: Possible solutions:

- syscall invokes kernel code, however, function call invokes user(application) code
- syscall switches the mode to privileged mode, however, the mode will remain in user mode when doing function call
- the transfer is made to a fixed location in memory when a syscall is made, however, the transfer is made to a caller specified location when a function call is made

6. (3 points) Match the terms in column A with the most appropriate definition from column B.

Column A	Column B
1. Synchronization	a. Piece of code that only one thread can execute at once
2. Mutual exclusion	b. Ensuring that only one thread does a particular thing at a time
3. Critical section	c. Isolating program faults to an address space
	d. Using atomic operations to ensure cooperation between threads

1 ⟷ d

2 ⟷ b

3 ⟷ a

Group C: Scheduling [15 Marks]

7. (3 points) For each of the scheduling algorithms listed below, indicate whether or not the algorithm is preemptive. If it is preemptive, describe briefly the conditions under which preemption occurs.

   (a) First come first served

   Solution: non-preemptive

   (b) Round robin

   Solution: preemptive. When the quantum time allocated for a process finishes.

   (c) Shortest Remaining Time Next

   Solution: preemptive. When a new process arrives and it has a shorter service time as compared to the running process.

8. (2 points) What happens if the time slice allocated in a Round Robin Scheduling is very large? And what happens if the time slice is very small?

   Solution: If time slice is very large, it results in FCFS scheduling. If time slice is too small, the processor through put is reduced, since more time is spent on context switching.

9. (10 points) Consider the below table that has information about three processes and their burst times. The CPU burst time is the time required by the process to execute on the processor with no interruption to perform I/O or any other blocking system call.

Process	Burst Time (Time Unit)
P1	7
P2	1
P3	16

(a) (5 points) Assume that all the three processes are ready for execution at time 0. Their order in the queue are as follows: P1, then P2, and finally P3. Explain why using shortest-process-next (SPN) is better than first come first served (FCFS) scheduling. Justify your answer by calculating the turnaround time.

Solution: FIFO:
P1 runs from 0 to 7 – turnaround time = 7
P2 runs from 7 to 8 – turnaround time = 8
P3 runs from 8 to 24 – turnaround time = 24
Average turnaround time = (7+8+24)/3 = 13

SPN
P2 runs from 0 to 1 – turnaround time = 1
P1 runs from 1 to 8 – turnaround time = 8
P3 runs from 8 to 24 – turnaround time = 24
Average turnaround time = (8+1+24)/3 = 11

Therefore, SPN yields a lower average turnaround time.

(b) (5 points) Assume that P1 and P3 are ready for execution, but P2 becomes ready after 2 time units. Can SPN still perform well? How can we modify the scheduling algorithm to still achieve an optimal execution order that minimizes the average turnaround time. What is the resulting execution schedule?

Solution: If SPN is used, the turnaround time will be similar to that of the FIFO. P1 runs from 0 to 7, P2 runs from 7 to 8, and finally P3 runs from 8 to 24.
The scheduling algorithm can be modified by preempting the running process if a shorter process is added to the system, or shortest remaining time next. In this case:
P1 runs from 0 to 2
P2 preempts P1 and runs from 2 to 3 – turnaround time =1
P1 resumes from 3 to 8 – turnaround time = 8
P3 runs from 8 to 24 – turnaround time = 24
Therefore, the average turnaround time is 11.

Group D: Memory Management [16 Marks]

10. (2 points) In class, we discussed three heuristics for memory placement when we are using variable sized memory allocation: first fit, best fit, and worst fit. Briefly describe the worst fit allocation strategy, and explain the motivation behind it.

> **Solution:** The worst fit heuristic specifies that the OS should allocate the largest free memory area or "hole" to satisfy a memory request. This results in the largest possible leftover hole after satisfying the memory request. The motivation behind worst fit is that such a large leftover hole would be more useful for satisfying future memory requests that the smaller leftover holes that we get with first fit or best fit.

11. (10 points) A virtual memory architecture has the following parameters:

    - 37 bit virtual addresses
    - 44 bit physical address
    - 16K (2^{14}) byte page size
    - each virtual page only has valid and dirty bits associated with it

    (a) (6 points) Show how a virtual address is translated into a physical address. Make sure to show the following:

    i. The number of bits allocated to page number and offset in the virtual address.
    ii. The number of bits allocated to page frame number and offset in the physical address.
    iii. The maximum number of entries the page table could hold.
    iv. The maximum size in bytes of the page table.

> **Solution:**
>
> 1. virtual address is divided into: page number: 23 bits and offset: 14 bits
> 2. virtual address is divided into: page frame number: 30 bits and offset: 14 bits
> 3. page table size: page width: 30 bits + 2 bits , number of entries: 2^{23}.
> 4. Therefore total size: $2^{23} * 2^5 = 2^{28}$ bits = 2^{25} bytes

(b) (2 points) The page tables for this memory system may become very large, especially if they are sparsely populated. Describe a technique for the memory system to store the tables more efficiently, and explain how this technique will improve the storage efficiency of the page tables.

Solution: Using multilevel page tables. Only the base level need to be in memory

(c) (2 points) Describe how the address translation will change is we add a TLB for this memory system. The number of rows of the TLB are 1024. Draw a diagram of the TLB, showing the size of each field in bits. Indicate how bits of the virtual address are used for input to the TLB, and describe the outputs from the TLB.

Solution: The TLB entry will contain: page num (23 bits), and corresponding frame number (30 bits) and validation and dirty bits. This is a total of 55 bits .

12. (2 points) What is the difference between internal and external memory fragmentation?

Solution: internal: (paging) the minimum unit of memory allocated to a process is a page. When less than a page is required, that memory is wasted
external: (dynamic relocation) all memory allocations must be contiguous. When there are blocks of unallocated space between allocations that are too small to be used, that memory is wasted

13. (2 points) In virtual memory systems, why is replacing a clean page faster than replacing a dirty page? A dirty page is one that has been modified while it is loaded in memory.

Solution: When the OS replaces a dirty page, it has to write this page to the swap disk. A clean page can be replaced without writing it to the swap disk.

14. (6 points) Consider a virtual memory system that uses paging. Virtual and physical addresses are both 32 bits long, and the page size is 4KB = 2^{12} bytes. A process P1 has the following page table. Frame numbers are given in hexadecimal notation (Hint: each hexadecimal digit represents 4 bits).

	Page Frame Number
0	0x0014E
1	0x03B65
2	0x00351
3	0x00875
4	0x06A3F

(a) For each of the following virtual addresses, write the physical address to which it maps. If the virtual address is not part of the address space of P1, write "NO TRANSLATION" instead. Use hexadecimal notation for the physical addresses.

i. 0x00003B65

Solution: 0x00875B65

ii. 0x00006A3F

Solution: NO TRANSLATION

iii. 0x00000FE6

Solution: 0014EFE6

(b) For each of the following physical addresses, write the virtual address that maps to it. If the physical address is not part of the physical memory assigned to P1, write "NO TRANSLATION" instead. Use hexadecimal notation for the virtual addresses.

i. 00351FFF

Solution: 00002FFF

ii. 03B65000

Solution: 00001000

iii. 000E3000

Solution: NO TRANSLATION

Group E: Input/Output Management and File System [6 Marks]

15. (4 points) Briefly describe the steps performed to read a block of data from the disk to the memory when using DMA controlled I/O.

Solution:

1. CPU sets the DMA controller registers with information about the data to transfer and where
2. CPU issues a command to the Disk controller to read the data and apply checksum
3. DMA controller issues a read request over the bus to the disk controller
4. Data is transferred to the memory
5. Disk controller acknowledges completing transfer of the data ? sent to DMA controller
6. DMA interrupts the CPU
7. OS starts and it does not have to copy data into memory as it is already there

Brief sol:

1. CPU programs the DMA controller by setting its registers so it knows what to transfer where.
2. The DMA controller initiates the transfer by issuing a read request to the disk controller.
3. The disk controller fetches the next word from its internal buffer and writes it to the memory.
4. When the write is finished, the disk controller sends an acknowledgement to the DMA controller
5. If there are more data to transfer (counter is greater than 0), then the DMA controller repeats steps 2 through 4. If all the data has been transferred, the DMA controller interrupts the CPU to let it know that the transfer is complete

16. (2 points) Briefly (in 2-3 sentences) explain the differences between a hard link and a soft link.

Solution: Hard links point to the same inode, while soft links simply contain the name of a directory entry. Hard links use reference counting. Soft links do not and may have problems with dangling references if the referenced file is moved or deleted. Soft links can span file systems, while hard links are limited to the same file system.

Group F: Creativity

17. (Bonus-Maximum 2 points) Describe what memory thrashing is in a creative way. For example, you can draw a picture, write a poem, ... A hilarious contribution is encouraged.