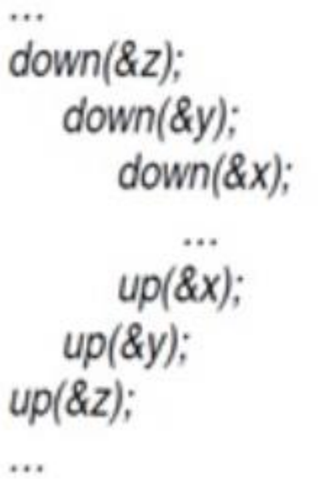
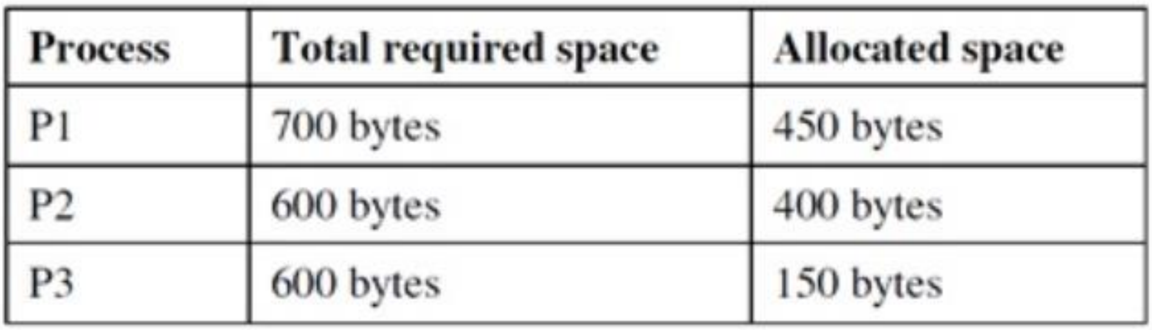
1.	(i) In UNIX file system, give an Algorithm for Conversion of a Path Name to an inode. (ii) List Problems in Original (Old) Unix File System. How Fast File System (FFS) overcomes. Give FFS Data Layout and analyze how do we Achieve Locality in FFS.
2.	Explain about structure of a regular file and bmap algorithm in detail. Given a disk-block size of 1 KB and block-pointer address value of 8 bytes, what is the largest file size (in bytes) that can be accessed using 10 direct addresses and one indirect block?
3.A.	List the system calls that are invoked (with their arguments), when you write and execute your own UNIX system program to implement "ls | wc -l". what if ls produces output faster than wc consumes it? what if ls is slower than wc? how does each command decide when to exit? what if reader didn't close the write end? what if writer didn't close the read end? how does the kernel know when to free the pipe buffer?
3.B.	How write system calls work. Explain with an algorithm. Where are the read and write attributes of an open file descriptor stored? Suppose a process wants to write a few bytes. Let's assume we want to write 100 bytes, starting with byte 2000 in the file. This will be expressed by the pair of system calls: seek(fd,2000); write(fd,buf,100); Let's also assume that each disk block is 1024 bytes. Illustrate how Writing may require new blocks to be allocated according to its internal structure, algorithms, and data structures.
4.A.	Explain Inode Life Cycle with ialloc(), iput() algorithms.
4.B.	How to build a distributed file system? In NFS distributed file system, Illustrate Handling Server Failure with Idempotent Operations for the three Types of Losses.
5.	How many levels does the UNIX scheduling algorithm include? What are they? In its low-level algorithm, how is the priority value for every process computed? What does each of the three components in the priority formula mean, respectively? How does the priority formula indicate that UNIX gives higher priority to processes that have used less CPU time in the recent past? Explain the reason by describing the computation process of the priority formula. Consider the following set of processes with CPU burst time of each of them is given in milliseconds. Process CPU burst time (ms) (time required by a process) P1 15 P2 5 P3 7 P4 10 Draw the Gantt chart for round robin scheduling where time quantum q = 4 milliseconds. Calculate the average waiting time and turnaround time.
6.	(i) List Classes of Real-Time Scheduling Algorithms and analyze them. (ii) Why per-process kernel stack? Explain Scheduler () in xv6, Give the summary of entries in xv6 PCB.
7.A.	Explain the function sched used by a process that wishes to relinquish the CPU. How does xv6 store process state? How does an OS keep time? Suppose a process goes to sleep and the system contains no process ready to run. What happens when the sleeping process does its context switch? Discuss proc.tf (trapframe)
7.B.	List and illustrate the tasks performed by the fork system call by giving an algorithm. After a process has exited, it may enter the state of being a ZOMBIE before disappearing from the system entirely. What is the purpose of the ZOMBIE state? What event causes a process to exit from ZOMBIE?
8.A.	Write a program that creates two child processes. Each created (child) process has to execute a separate executable program. Make the parent process wait until both child processes are completed, before being terminated. List the the actions performed by the exit() system call by giving the algorithm.
8.B.	Explain the three interrelated issues that involves Scheduling on a multiprocessor. In xv6, explain Scheduling Parameters and Controlling Process Priorities using swtch(), yield(), sched(). Discuss mycpu and myproc.

9.	A computer system uses non-segmented 128-bit virtual addresses mapping pages of size 8 kbytes with 256 Mbytes of physical memory. Show why a multi-level page table would not be effective here. Give and analyze Multi-level Page Table Control Flow algorithm.
10.	Explain Belady's anomaly with an example. Define trashing. Assume that memory contains only three frames. Initially all three frames are empty. The page reference string is 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1. Find how many page faults occurs using FIFO, Optimal Page replacement, and Least Recently Used algorithms. Pictorially show which pages are replaced.
11.A.	(i) Explain Completely-Fair-Scheduler (CFS). Is the Linux scheduling policy adequate for hard real-time systems? Explain. (ii) If one thread forks, is the entire process copied, or is the new process single-threaded?
11.B.	(i) In Address translation, explain static and dynamic binding. Differentiate Static vs. Dynamic Linking. (ii) Give your version of the calloc() call using system calls to allocate the swap space for the process to be swapped out. You can descript it with a C-like algorithm
12.A.	In xv6, explain Assembly bootstrap and C bootstrap with respective bootasm.S and bootmain.c
12.B.	Explain the role of page map table. For a given virtual address in a binary number, write and analyze paging control flow algorithm.
13.	Four processes are concurrently running. These processes compete for two reusable resources. Each process needs two resources to complete, but it can request the resources one at a time. How many different ways can deadlock occur? Compare the IPC functionality provided by pipes, FIFO's and message queues.
14.	Write a program to demonstrate deadlock using condition variable. How do condition variables avoid the lost wakeup problem? Give pseudocode to producer consumer problem using condition variables with multiple producer threads and single consumer thread.
15.A.	Many systems classify library functions as thread-safe or thread-unsafe. What causes a function to be unsafe for use by a multithreaded application? Implement Concurrent Hash Table that uses a lock per hash bucket each of which is represented by a list.
15.B.	Write programs to compare the data-transfer speed using shared memory and messages. The programs for shared memory should include semaphores to synchronize completion of reads and writes..
16.A.	(i) How is a Thread Create? Why is communication and data sharing among threads faster than processes? Illustrate the reasons. if the primary thread of a process exits, is it possible for the other threads of this primary thread to continue running? (ii) The Initial values of semaphores x, y, and z are 1, 5, and 10, respectively. At the most, how many processes may be awaiting each of these semaphores, with respect to the following pseudo code?



```
...
down(&z);
    down(&y);
        down(&x);
            ...
        up(&x);
    up(&y);
up(&z);
...
```

16.B.	(i) Three processes are concurrently running. There are three type R1 resources, two type R2 resources, and two type R3 resources. These processes are currently holding one R1, two R2, and one R1 resources, respectively. If the following requests arrived in the order below, would the system be safe? • P1 requests one R2 • P2 requests one R1 and one R3 • P3 requests one R1. (ii) Four processes are going to use a common buffer the size of 1500 bytes. The following table shows the maximum and allocated requirements of the first three processes. A process cannot finish unless its complete requirements are allocated. • If the fourth process enters with a total requirement of 600 bytes and it immediately requests 250 bytes, is this request safe (with respect to deadlock)? If the fourth process enters with a total requirement of 600 bytes and it immediately requests 350 bytes, is this request safe?



Process	Total required space	Allocated space
P1	700 bytes	450 bytes
P2	600 bytes	400 bytes
P3	600 bytes	150 bytes