Operating system CANNOT merge I/O devices and files into a combined file because of the similarity of system calls for each.

2) RMI is similar to RPC

The single **benefit of a thread pool** is to control the number of threads is **NOT** true.

All contemporary operating systems support kernel threads.

Race conditions are prevented by locks

Monitors ARE practiced in the modern programming languages.

If a **resource-allocation graph** has a cycle, it does NOT necessarily mean the system must be in a deadlocked state.

Ordering resources and requiring the resources to be acquired in order prevents the circular wait from occurring and therefore prevents deadlock from occurring: false

The banker's algorithm IS useful in a system with multiple instances of a resource type

A **non-preemptive scheduling** a process cannot be interrupted once it starts its execution, it will terminate only when it ends while in **preemptive scheduling** a process can be interrupted by another process in mid of its execution also.

LWP (Light Weight Process): is an intermediate data structure between user and kernel threads, which appears to be a virtual processor on which process can schedule user thread to run on a kernel thread

Gant Chart: Is a way to draw scheduled processes such as **priority scheduling.**

Spinlock: The Process "spins" while waiting for a lock. Spinlocks have the advantage that no context switch is required when a process must wait on a lock, and a context switch may take considerable time.

Semaphore: Is a software based synchronization tool used against the critical section problem.

Deadlock: Is a situation where two or more processes are waiting indefinitely for an event that can be caused only by one of the waiting processes.

Critical Section: Is a section of code that a process needs to execute, in which the process may be changing common variables, updating a table, or writing a file.

Throughput: Is measure of work for the number of processes that are completed per cpu time unit.

Long-term scheduler (or job scheduler) –selects which processes should be brought into the ready queue

Short-term scheduler (or CPU scheduler) – selects which process should be executed next and allocates CPU

Midterm Scheduler Sometimes it can be advantage to remove process from memory and thus decrease the degree of multi-programming. This scheme is called swapping

Process P0 and P1 set to 1: The implementation of a semaphore with a waiting queue may result in a situation where two or more processes are waiting indefinitely for an event that can be caused only by one of the waiting processes. The event in question is the execution of a signal() operation. When such a state is reached, these processes are said to be deadlocked.

Given the definition of a **resource-allocation graph**, it can be shown that, if the graph contains no cycles, then no process in the system is deadlocked. If the graph does contain a cycle, then a deadlock may exist. If each resource type has exactly one instance, then a cycle implies that a deadlock has occurred. If the cycle involves only a set of resource types, each of which has only a single instance, then a deadlock has occurred. Each process involved in the cycle is deadlocked. In this case, a cycle in the graph is both a necessary and a sufficient condition for the existence of deadlock.

Page Number: If the size of the logical address space is 2^m, and a page size is 2^n bytes, then the high-order $m-n$ bits of a logical address designate the page number, and the n low-order bits designate the page offset.

Paging partitions: M = Size of memory, N = Size of page.

First fit: Allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or at the location

Page Number	Page Offset
p	d
m - n	n

where the previous first-fit search ended. We can stop searching as soon as we find a free hole that is large enough.

Best fit: Allocate the smallest hole that is big enough. We must search the entire list, unless the list is ordered by size. This strategy produces the smallest leftover hole.

Worst fit: Allocate the largest hole. Again, we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.

Point-to-Point and Broadcast packet type?

: It maps many user-level threads to one kernel thread. The entire system may block makes a block system call.

One to one: Each user-level thread has one corresponding kernel thread. The only drawback is that creating a user thread requires creating the corresponding kernel thread

Many to many: Multiplexes many user-level threads to a smaller or equal number of kernel threads. User can create as many threads as they want

The major difficulty in designing a **layered operating system** approach is **appropriately defining the various layers**

A **boot block** _____. Is typically only knows the location and length of the rest of the bootstrap program

Microkernels use _____ for communication. Is message passing

The UNIX fork() system call creates a new process. What is the equivalent system call in WINDOWS? **CreateProcess ()**

All access to POSIX shared memory requires a system call: **False**

Cancellation points are associated with _____ cancellation. **Deferred**

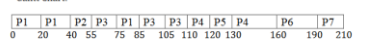The _____ model allows a user-level thread to be bound to one kernel thread. two-level

Throughput is the number of processes that are completed per time unit.

The **compile-time** and **load-time** address-binding methods generate identical logical and physical addresses. However, the **execution-time** address-binding scheme results in differing logical and physical addresses. In this case, we usually refer to the logical address as a virtual address

What is an **Operating System?: resource allocator, Control program**

By **"Multilevel Queue Fixed priority"** scheduling algorithm, draw the CPU scheduling Gantt chart (5 pts) and complete the table for the given processes information. (10 pts)

	Process	Burst time	Arriving time	Algorithm
■ 1st Foreground	P1	50	0.0	RR interval:20
	P2	15	30.0	(RR is a non-preemptive
	P3	45	30.0	algorithm)
■ 2nd Foreground	P4	40	0.0	SJF Preemptive
	P5	10	120.0	
■ Background	P6	30	60.0	FCFS
	P7	20	130.0	

Gantt Chart:

P1	P1	P2	P3	P1	P3	P3	P4	P5	P4		P6		P7
0	20	40	55	75	85	105	110	120	130		160	190	210

Scheduling criteria:

	P1	P2	P3	P4	P5	P6	P7
Waiting time	35	10	35	120	0	100	60
Turnaround time	85	25	80	160	10	130	80
Response time	0	10	25	110	10	100	60

	Process	Burst time	Arriving time	Algorithm
■ 1st Foreground	P1	50	0.0	RR interval:20
	P2	15	30.0	(RR is a non-preemptive
	P3	45	30.0	algorithm)
■ 2nd Foreground	P4	40	0.0	SJF Preemptive
	P5	10	120.0	
■ Background	P6	30	60.0	FCFS
	P7	20	130.0	

Gantt Chart:

P1	P1	P2	P3	P1	P3	P3	P4	P5	P4		P6		P7
0	20	40	55	75	85	105	110	120	130		160	190	210

Scheduling criteria:

	P1	P2	P3	P4	P5	P6	P7
Waiting time	35	10	35	120	0	100	60
Turnaround time	85	25	80	160	10	130	80
Response time	0	10	25	110	10	100	60

Resource-allocation graph with a cycle but no deadlock.
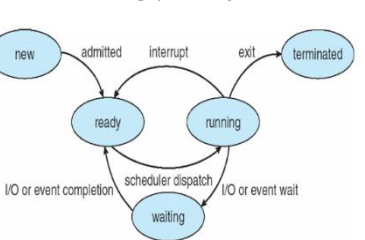
Producer View
```
while (true) {
/* produce an item and put in next Produced*/
    while (count == BUFFER_SIZE)
        ; // do nothing

    buffer [in] = nextProduced;
    in = (in + 1) % BUFFER_SIZE;
    count++;
}
```

Consumer View
```
while (true) {
    while (count == 0)
        ; // do nothing

    nextConsumed= buffer[out];
    out = (out + 1) % BUFFER_SIZE;
    count--;
    /* consume the item in next Consumed */
}
```

Time	Action	waiting:0	count:0	in:1	out:1
1	Producer produces 2 jobs	0	2	3	1
2	Consumer consumes 1 job	0	1	3	2
3	Producer produces 3 jobs	0	4	2	2
4	Consumer consumes 3 jobs	0	1	2	1
5	Consumer consumes 3 jobs	1	0	2	2

Which of the following is true **of cooperative scheduling**? A process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state

Paging: The basic method for implementing paging involves breaking physical memory into fixed-sized blocks called frames and breaking logical memory into blocks of the same size called pages. When a process is to be executed, its pages are loaded into any available memory frames from their source (a file system or backing store)

The most common systems use **symmetric multiprocessing (SMP)**, in which each processor performs all tasks within the operating system. SMP means that all processors are peers; no boss – worker relationship exists between processors.

The **multiple-processor systems** in use today are of two types. Some systems use **asymmetric multiprocessing**, in which each processor is assigned a specific task.

One important principle is the separation of **policy** from **mechanism**. Mechanisms determine how to do something; policies determine what will be done.

The main advantage of the **layered approach** is simplicity of construction and debugging. The layers are selected so that each uses functions and services of only lower-level layers.

The **major difficulty with the layered approach** involves appropriately defining the various layers. Because a layer can use only lower-level layers, careful planning is necessary

PCB STUFF

Process state. The state may be new, ready, running, and waiting, halted, and so on.

Program counter. The counter indicates the address of the next instruction to be executed for this process.

CPU registers. Include accumulators, index registers, stack pointers, and general-purpose registers, plus any condition-code

CPU-scheduling information. This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters.

Memory-management information. This information may include such information as the value of the base and limit registers, the page tables, or the segment tables, depending on the memory system used by the operating system).

Accounting information. This information includes the amount of CPU and real time used, time limits, account numbers, job or process numbers, and so on.

I/O status information. This information includes the list of I/O devices allocated to the process, a list of open files... When a **context switch** occurs, the kernel saves the context of the old process in its PCB and loads the saved context of the new process scheduled to run. Context-switch time is pure overhead, because the system does no useful work while switching

A socket is defined as an endpoint for communication. A pair of processes communicating over a network employ a pair of sockets—one for each process. A socket is identified by an IP address concatenated with a port number.

Turnaround time. Is how long it takes to execute a process. The interval from the time of submission of a process to the time of completion is the turnaround time. **Turnaround time** is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O.

Waiting time is the sum of the periods spent waiting in the ready queue.

Response time. Is the time it takes to start responding, not the time it takes to output the response. The **turnaround** time is generally limited by the speed of the output device.

The multilevel feedback queue scheduling algorithm allows a process to move between queues. The idea is to separate processes according to the characteristics of their CPU bursts. If a process uses too much CPU time, it will be moved to a lower-priority queue. This scheme leaves I/O-bound and interactive processes in the higher-priority queues. In addition, a process that waits too long in a lower-priority queue may be moved to a higher-priority queue. This form of aging **prevents starvation**

Check for safety by using a **cycle-detection algorithm**. An algorithm for detecting a cycle in this graph requires an order of n^2 operations, where n is the number of processes in the system.

Distributed system is collection of loosely coupled processors interconnected by a communications network

Processors variously called **nodes, computers, machines, hosts,** | **Site** is location of the processor | Generally a **server** has a resource a **client** node at a different site wants to use

Local-Area Network (**LAN**) – designed to cover small geographical area =>

Multiple topologies like **star** or **ring**, | **Speeds** from 1Mb per second (Appletalk, bluetooth) to 40 Gbps for fastest Ethernet over twisted pair copper or optical fibre | **Consists** of multiple computers (mainframes through mobile devices), peripherals (printers, storage arrays), routers (specialized network communication processors) providing access to other networks | **Ethernet** most common way to construct LANs | **Multiaccess** bus-based | **Defined** by standard IEEE 802.3 | Wireless spectrum (**WiFi**) increasingly used for networking | I.e. IEEE 802.11g standard implemented at 54 Mbps

Wide-Area Network (WAN) – links geographically separated sites **Point-to-point** connections over long-haul lines (often leased from a phone company)

Speeds | **T1** link is 1.544 Megabits per second | **T3** is 28 x T1s = 45 Mbps | **OC**-12 is 622 Mbps

Layer 1: Physical layer – handles the mechanical and electrical details of the physical transmission of a bit stream

Layer 2: Data-link layer – handles the *frames*, or fixed-length parts of packets, including any error detection and recovery that occurred in the physical layer
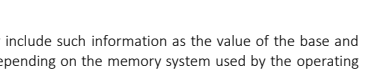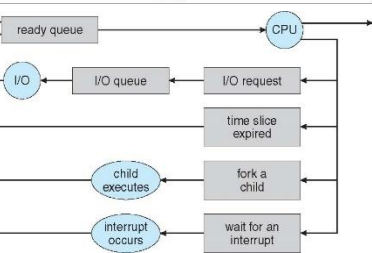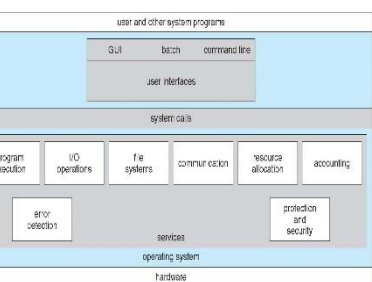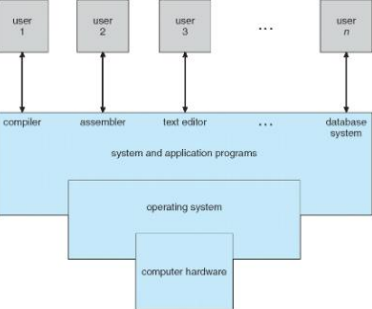
Layer 3: Network layer – provides connections and routes packets in the communication network, including handling the address of outgoing packets, decoding the address of incoming packets, and maintaining routing information for proper response to changing load levels

Layer 4: Transport layer – responsible for low-level network access and for message transfer between clients, including partitioning messages into packets, maintaining packet order, controlling flow, and generating physical addresses

Layer 5: Session layer – implements sessions, or process-to-process communications protocols

Layer 6: Presentation layer – resolves the differences in formats among the various sites in the network, including character conversions, and half duplex/full duplex (echoing)

Layer 7: Application layer – interacts directly with the users, deals with file transfer, remote-login protocols and electronic mail, as well as schemas for distributed databases