

# Operating Structures

**Q1:** List at least three operating system services that are useful to users.

**1. User Interface:**

**Def:** The way of interaction of User with Computer is called User Interface.

**2. Program execution:**

**Def:** process by which a computer reads and acts on the instructions of a computer program.

**3. I/O operations**

**Def:** input and output actions.

**Q2:** List at least three operating system functions that maintain efficient operation of the system.

**1. Resource allocation**

**Def:** Assignment of resources among the Processes.

**2. Accounting**

**Def:** record-keeping and tracking of user activities on computer network.

**3. Protection and security**

**Protection:** - any mechanism for controlling access of processes.

**Security:** - defense of the system against internal and external attacks.

**Q3:** What are the two different approaches for providing a user interface?

1. CLI (command-line interface)

2. GUI (graphical user interface)

**Q4:** What is a system call?

An interface to the services made available by an operating system.

## **Q5: What is an API?**

An interface that specifies a set of functions that are available to an application programmer, including the parameters that are passed to each function and the return values the programmer can expect.

## **Q6: What kernel data structure can be used for one technique of passing parameters to system calls?**

### **Stack**

**Def:** A stack represents a sequence of objects or elements in a linear data structure format. (It contains temporary data such as local variables, return addresses, and function parameters)

## **Q7: List at least three of the major categories of system calls.**

### **1. process control**

**Def:** the ability to monitor and adjust a process to give a desired output.

### **2. file manipulation**

**Def:** File manipulation functions handle creating, opening, and saving documents.

### **3. device manipulation**

**Def:** requesting the Devices for acquire the desired output.

## **Q8: A program that has been loaded and executing is called a \_\_\_\_.**

### **Process**

## **Q9: What part of the operating system makes the decision with regards to which job will run?**

### **command interpreter**

**Q10:** List at least three of the categories of system programs.

**1. File management**

**Def:** For File Management a file manager is used which provides a user interface to manage files and folders.

**2. Status information**

**Def:** It provide required data on the current or past status of the system.

**3. File modification**

**Def:** changing or asking permissions for file and folder access.

**Q11:** True or False? The view most users see of the operating system is defined by application and system programs rather than system calls.

**True**

**Q12:** What are the two basic goal groups that must be considered when designing an operating system?

**1. User goals**

**2. System goals**

**Q13:** What is the difference between policy and mechanism?

**Mechanisms** determine how to do something,  
**policies** determine what will be done.

**Q14:** List at least three different ways for structuring an operating system.

**1. Simple structure** (MS-DOS & early UNIX)

**2. Layered approach** (layer 0 is hardware, layer N is user interface)

**3. Hybrid structure** (Mac OS X, uses Mach microkernel and the BSD UNIX kernel)

**\*5. Microkernel** (Tru64 UNIX)

**\*4. Modules** (Solaris, Linux, Windows)

**Q15:** List at least two different hybrid operating systems.

1. Mac OS X
2. iOS
- \*3. Android

**Q16:** What are the two devices that run the iOS operating system?

1. Apple iPhone
2. Apple iPad

**Q17:** What technique do microkernels use to communicate between services?

message passing

**Q18:** Provide an example of an operating system that uses the simple structure.

MS-DOS

**Q19:** True or False? Performance tuning is a type of debugging.

True

**Q20:** True or False? DTrace is available for Windows systems.

False, it has been added to Mac OS X and FreeBSD

**Q21:** Name two activities the operating system is responsible for in connection with disk management.

1. Disk I/O
2. Disk space/capacity

**Q22:** Of the following 5 forms of storage, rank them from fastest to slowest in terms of access time:

**Largest to Smallest (Size)**

1. magnetic disk (Non-Volatile)
2. solid state disk (Non-Volatile)
3. main memory (Volatile)
4. Cache (Volatile)
5. Registers (Volatile)

**fastest to slowest (Speed)**

1. registers
2. cache
3. main memory
4. solid state disk
5. magnetic disk

**Q23:** What does the term SYSGEN refer to?

**system generation:** the process of generating an instance of an operating system for each individual computer site/architecture.

**Q24:** What is the name of the small piece of code that locates the kernel and loads it into main memory?

bootstrap program / bootstrap loader

# Processes

**Q1:** What are the four components of a process?

1. **Stack** (contains temporary data such as local variables, return addresses, and function parameters)
2. **Heap** (memory that is dynamically allocated during process runtime)
3. **Data Section** (contains global variables)
4. **Text Section** (program code)

**Q2:** Provide at least three possible states a process may be in.

1. **New** (process creation)
2. **Running** (instructions are being executed)
3. **Waiting** (The process is waiting for some event to occur)
- \*4. **Ready** (process is waiting to be assigned to a processor)
- \*5. **Terminated** (process has finished execution)

**Q3:** What is a Process Control Block (PCB)?

A representation of the process in the operating system, containing the  
Process state,  
Program counter,  
CPU registers,  
CPU-scheduling information, and  
Memory management information.

**Q4:** What is another term for process?

**Job**

**Q5:** True or False? Most operating systems allow a process to have multiple threads.

**True**

**Q6:** What is the role of the process scheduler?

It selects an available process/Job (possibly from a set of several available process) for program execution on the CPU.

**Q7:** What is the degree of multiprogramming?

It is the number of processes in memory.

**Q8:** What is the term that describes saving the state of one process, and restoring the state of another?

**Context Switch**

**Q9:** What is a process identifier (PID)?

A number, typically an integer, that uniquely identifies a process on a system.

**Q10:** What system call creates a process on UNIX systems?

**fork()**

**Q11:** What system call creates a process on Windows systems?

**CreateProcess()**

**Q12:** What system call terminates a process on UNIX systems?

**exit()**

**Q13:** What is the name of the process that UNIX and Linux systems assign as the new parent of orphan processes?

The **init** process is assigned as the parent of orphan processes.

**Q14:** What are the two fundamental models of interprocess communication?

1. **shared memory**

**Def:** An area of memory shared among the processes that wish to communicate.

2. **message passing**

**Def:** processes communicate with each other by establish a communication link between them and exchange messages via send/receive function.

**Q15:** What are the two system calls used with message-passing systems?

1. **send()**

2. **receive()**

**Q16:** True or False? Message passing is typically faster than shared memory?

**False**



**Q17:** How must share memory behave for a rendezvous to occur?

Both the send() and receive() system calls must be blocked for a rendezvous to occur.

**Q18:** What system call is used to create a POSIX shared memory object?

shm\_open()

**Q19:** What term does Mach use to describe mailboxes?

Ports

**Q20:** What system call does Mach use to create a new mailbox?

port\_allocate()

**Q21:** What term does Windows use to name its message passing facility?

advanced local procedure call (ALPC)

**Q22:** Provide at least two types of communication mechanisms in client-server systems.

1. **Sockets** (endpoints for communications)

2. **Remote Procedure Calls** (RPC) (a powerful technique for constructing distributed, client-server based applications.)

\*3. Pipes (a conduit to allow two processes to communicate; implemented in early UNIX systems)

**Q23:** TCP sockets are (a) connection-oriented, or (b) connection-less?

(a) connection-oriented

**Q24:** UDP sockets are (a) connection-oriented, or (b) connection-less?

(b) connection-less

**Q25:** \_\_\_\_\_ abstract procedure calls for use between systems with network connections.

Remote Procedure Calls (RPC)

**Q26:** What is parameter marshalling?

Packaging the parameters into a form that can be transmitted over a network.

**Q27:** What are the two types of pipes?

1. Ordinary Pipes
2. Named Pipes

# Thread

**Q1:** How many threads does a traditional, heavyweight process have?

A single thread of control

**Q2:** Provide at least three benefits of multithreaded programming.

**1. Responsiveness**

**Def:** multithreaded programming may allow continued execution if part of process is blocked

**2. Resource sharing**

**Def:** threads share resources of process, easier than shared memory or message passing

**3. Economy**

**Def:** cheaper than process creation, thread switching lower overhead than context switching

\*4. Scalability

**Q3:** True or False? Concurrency is only possible with parallelism.

**False**, it is possible to have concurrency without parallelism.

**Q4:** True or False? Amdahl's Law addresses the disproportionate effect of the serial portion of a program.

**True**

**Q5:** List at least three challenges when designing program for multicore systems.

**1. Identifying tasks**

**2. Balance**

**3. Data splitting**

\*4. Data dependency

\*5. Testing and debugging

**Q6:** What are the two general types of parallelism?

1. Data parallelism
2. Task parallelism

**Q7:** List the three common ways of mapping user threads to kernel threads.

**1. Many-to-one**

**Def:** Many user-level threads mapped to single kernel thread.

**2. One-to-one**

**Def:** Each user-level thread maps to kernel thread.

**3. Many-to-many**

**Def:** Allows many user level threads to be mapped to many kernel threads.

**Q8:** True or False? Only Linux and Windows implement the one-to-one model.

True

**Q9:** What are the two approaches for implementing a thread library?

1. Provide the library in the user space with no kernel support.
2. Implement a kernel-level library supported directly by the operating system.

**Q10:** What are the three main thread libraries in use?

1. POSIX Pthreads
2. Windows
3. Java

**Q11:** True or False? PThreads is typically only implemented on UNIX-like systems.

True

**Q12:** True or False? PThreads is only a specification, not an implementation.

True

**Q13:** What is the PThread API for creating a thread?

`pthread_create()`

**Q14:** What is the Windows API for creating a thread?

`CreateThread()`

**Q15:** What Java method is used for allocating and initializing a new thread in the JVM?

`start()`

**Q16:** Provide at least two techniques for supporting implicit threading

### 1. Thread Pools

**Def:** Create a number of threads in a pool where they await work.

### 2. OpenMP

**Def:** Provides support for parallel programming in shared-memory environments.  
(It creates as many threads as there are cores)

### 3. Grand Central Dispatch (GDP)

**Def:** Allows identification of parallel sections and manages most of the details of threading

**Q17:** True or False? Grand Central Dispatch only works for Apple's Mac OS X and iOS operating systems.

True

**Q18:** True or False? The semantics of the fork() system call can vary on multithreaded systems.

True

**Q19:** What are the two scenarios for cancelling a target thread?

1. **Asynchronous Cancellation** (One thread immediately terminates the target thread)
2. **Deferred cancellation** (The target thread periodically checks whether it should terminate)

**Q20:** What is the PThreads API for thread cancellation?

pthread\_cancel()

**Q21:** True or False? Windows threads provide both user and kernel stacks.

True

**Q22:** What term does Linux use to refer to a process or a thread?

Task

# Process Synchronization

**Q1:** If the current value of counter = 5, what are its possible values if the producer and consumer processes run concurrently?

The possible values, if both the consumer and producer processes run concurrently, are 5, 6, or 7.

**Q2:** What is the term for describing the situation where shared data may be manipulated concurrently and the outcome of the execution depends upon the order of access?

race condition

**Q3:** What is the term used to describe the segment of code where shared data is accessed and possibly manipulated?

critical section

**Q4:** What are the three requirements a solution to the critical-section problem must satisfy?

1. Mutual exclusion
2. Progress
3. Bounded waiting

**Q5:** True or False? A non-preemptive kernel is essentially free from race conditions.

True

**Q6:** True or False? There are no guarantees Peterson's solution works correctly on modern computer architectures.

True

**Q7:** True or False? All solutions to the critical section problem are based on the premise of locking.

True\* (All modern solutions)

**Q8:** What are the two general hardware instructions that can be performed atomically?

1. test\_and\_set()
2. compare\_and\_swap()

**Q9:** What are the two functions used with mutex locks?

1. test\_and\_set()
2. compare\_and\_swap()

**Q10:** True or False? A spinlock is a type of mutex lock.

True

**Q11:** True or False? Semaphores can provide the same functionality as mutex locks.

True

**Q12:** What are the two operations that can be performed on a semaphore?

1. wait()
2. signal()



**Q13:** True or False? A binary semaphore is functionally equivalent to a mutex lock.

True

**Q14:** What are the names of the two processes associated with the bounded-buffer problem?

1. producer
2. consumer

**Q15:** How many writers may concurrently share the database with the readers-writers problem ?

Only one writer may have access to the database at a time.

**Q16:** What is the problem if all philosophers simultaneously pick up their left chopstick?

If all philosophers pick up the left chopstick, it will leave no right chopstick for them to pickup, creating a deadlock while they all indefinitely wait for a right chopstick to become available.

**Q17:** What are the two operations that can be performed on a condition variable?

1. wait()
2. signal()

**Q18:** Name at least one modern programming language that has incorporated the idea of a monitor.

1. Java

\*2. C#

**Q19:** What are the two states of a Windows dispatcher object?

1. signalled
2. Non-signalled

**Q20:** What is available in Linux for updating an integer variable without having to use locks?

atomic integers

**Q21:** True/ False? Linux uses spinlocks for both single & multiple processor systems

False; for single processor systems Linux uses kernel preemption.

**Q22:** What are the Pthreads operations for locking and unlocking a mutex lock?

pthread\_mutex\_lock() and pthread\_mutex\_unlock()

**Q23:** Provide at least one alternative to mutex locks, semaphores, reader writer locks, and monitors that provide support for concurrent programming.

1. transactional memory

\*2. OpenMP

**Q24:** True or False? The system model for deadlocks first requires a process request a resource, then use the resource, and finally release the resource

False; The system model for **normal mode** of operation requires a process request a resource, use the resource, and finally release the resource.

**Q25:** What are the four necessary conditions for characterizing deadlock?

1. Mutual exclusion
2. Hold and wait
3. No preemption
4. Circular wait

**Q26:** Describe one strategy for dealing with deadlocks?

1. Deadlock Prevention/Avoidance - use a protocol to prevent or avoid deadlocks, effectively ensuring that the system will NEVER enter a deadlocked state.

\*2. Deadlock detection and recovery

\*3. Ignore the potential for deadlocks and rely on means of manual recovery if a deadlock happens.

**Q27:** What is the only reasonable condition that can be used to prevent deadlocks from occurring?

Prevention of any one of the four necessary conditions that create a deadlocked process.

# CPU Scheduling

**Q1:** What are the two bursts that CPU schedulers are designed around?

- 1. CPU burst (The state of process under execution is called **CPU burst**)
- 2. I/O burst (The state of process under I/O request & its handling is called **I/O burst**)

**Q2:** True or False? Under preemptive scheduling, when a process switch from the running to the ready state, it may lose control of the CPU.

**False;** this is an example of cooperative or nonpreemptive scheduling.

**Q3:** List at least three different criteria for designing a CPU scheduling algorithm.

- 1. CPU utilization (keep the CPU as busy as possible)
- 2. Throughput (Number of processes that complete their execution per time unit)
- 3. Turnaround time (Amount of time to execute a particular process)
- \*4. Waiting time (Amount of time a process has been waiting in the ready queue)
- \*5. Response time (amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment))
- ....

**Q4:** What scheduling algorithm assigns the CPU to the process with the highest priority?

**Priority Scheduling** (A priority number (integer) is associated with each process)

**Q5:** True or False? The multilevel feedback queue scheduling algorithm allows processes to migrate between different queues.

**True**

**Q6:** What scheduling algorithm assigns the CPU to the process that first requested it?

**FIFO** (first in, first out) **Scheduling**

**Q7:** What scheduling algorithm assigns the CPU to a process for only its time slice (or time quantum?)

**Round-Robin Scheduling**

(Each process gets a small unit of CPU's time usually 10-100 milliseconds. After this time has elapsed, the process is preempted and added to the end of the ready queue)

**Q8:** What scheduling algorithm assigns the CPU to the process with the shortest burst?

**SJF** (shortest job first) **Scheduling**

**Q9:** What are the two types of contention scope for thread scheduling?

- 1. **process-contention scope** (Also called: Process local scheduling on Unbounded thread)
- 2. **system-contention scope** (Also called: Process global scheduling on bounded thread)

**Q10:** What are the two general hardware instructions that can be performed atomically?

- 1. **test\_and\_set()**
- 2. **compare\_and\_swap()**

**Q11:** What is more common on current systems, asymmetric or symmetric multiprocessing?

**symmetric multiprocessing (SMP)**

## Q12: What are the two forms of processor affinity?

1. **soft affinity** (It is not guaranteed that process will run on desired Processor)
2. **hard affinity** (Process is bound to run on desired processor)

## Q13: What are the two general approaches for load balancing?

1. **push migration** (periodic task checks load on each processor, and if load found, it pushes task from overloaded CPU to other CPUs)
2. **pull migration** (idle processors pulls waiting task from busy processor)

## Q14: What are the two ways to multithread a processing core?

1. **coarse-grained** (Coarse-grained systems consist of fewer but larger components)
2. **fine-grained** (Fine-grained systems consist of more but smaller components)

## Q15: What are the two general types of real-time scheduling?

1. **Soft real-time systems** (no guarantee that to when critical real-time process will be scheduled)
2. **Hard real-time systems** (task must be serviced by its deadline)

## Q16: What real-time scheduling algorithm uses deadline as its scheduling criteria?

**Hard real-time systems**

## Q17: What real-time scheduling algorithm is used for scheduling periodic tasks with static priorities?

- rate-monotonic** (In RM priorities are assigned according to time period)  
(Rate monotonic is a preemptive algorithm which means if a task with shorter period comes during execution it will gain a higher priority and can block or preemptive currently running tasks.)

**Q18:** What is the name of the default scheduling algorithm for current Linux systems?

**Completely Fair Scheduler (CFS)**

**Q19:** True or False? A Windows thread is assigned both a priority class and a relative priority within that class.

**True**

**Q20:** If a thread on a Solaris system exhausts its time quantum, will it later be assigned a higher or lower priority?

**lowered**

**Q21:** True or False? Deterministic modelling and simulations are similar strategies for evaluating scheduling algorithms

**True**

# Main Memory

**Q1:** What two registers can be used to provide a simple form of memory protection?

1. Relocation register
2. Limit register

**Q2:** List the three different times at which address binding may occur:

1. **Compile Time** (If the compiler is responsible for performing address binding, then it is called compile-time address binding. It will be done before loading the program into memory)
2. **Load Time** (This type of address binding will be done by the OS memory manager. It will be done after loading the program into memory)
3. **Execution Time** (The address binding is done by the processor at the time of program execution. The program will be kept on changing the locations in memory until the time of program execution)

**Q3:** True or False? An address generated by the CPU is also referred to as a physical address.

**False**, an address generated by the CPU is a logical address. An address seen by the memory unit is a physical address.

**Q4:** What is the hardware device that maps virtual to physical addresses?

**Memory-Management Unit**

(It is a computer hardware unit having all memory references passed through itself, primarily performing the translation of virtual memory addresses to physical addresses)

{

**backing store** is a commonly fast disk that must be large enough to accommodate copies of all memory images for all users, and it must provide direct access to these memory images.

}



## Q5: True or False? Mobile systems typically use swapping.

**False**, instead of swapping, when free memory falls below a certain threshold, Apple's iOS and Android asks applications to voluntarily relinquish memory.

{  
(Read only data (code) are removed from the system and later reloaded from flash memory if necessary, and the data that has been modified (stack) are never removed)

(Swapping is moving processes temporarily out of memory to a backing store)

}

## Q6: What are the three strategies for selecting a free hole from the set of available holes?

1. **First fit** (allocate the first hole that is big enough)
2. **Best fit** (allocate the smallest hole that is big enough)
3. **Worst fit** (allocate the largest hole)

## Q7: What are the two forms of fragmentation?

### 1. External fragmentation

In this type processes are loaded and removed from memory, the free memory space is broken into little pieces and exists when there isn't enough total memory space to satisfy a request

### 2. Internal fragmentation

In this type the unused memory that is internal to a partition, breaks the physical memory into fixed-size blocks (usually of base 2) and allocate memory in units based on block size. The memory allocated to a process may be slightly larger than the requested memory

**NOTE:** Both internal fragmentation and external fragmentation are phenomena where memory is wasted. Internal fragmentation occurs in fixed size memory allocation while external fragmentation occurs in dynamic memory allocation.

## Q8: List at least two possible parts of a program that may be assigned separate segments.

1. **Code**
2. **Global variables**
- \*3. The heap
- \*4. Stacks for each thread
- \*5. Standard C Library

**Q9:** What are the two parts of an address generated by the CPU?

1. A page number
2. An offset

**Q10:** What does each entry in the page table contain?

The base address of each page in physical memory.

**Q11:** True or False? Fragmentation can still occur in paging systems.

**True,** Internal fragmentation can still occur. External fragmentation cannot

**Q12:** What is the term that describes when a page number is not present in the TLB?

**TLB miss.** A memory reference to the page table is made, and the page and frame number are added to the Translation Look-aside Buffer (TLB)

**NOTE:** TLB is a memory cache used to minimise the time taken to reach the location of a user's memory

**Q13:** If a page offset is 13 bits, how large (in bytes) is the page?

$2^{13}$  bytes

**Q14:** True or False? IA-32 address translation involves both paging and segmentation.

**True**

## **Q15:** What is an alternative to hierarchical paging for large (>32 bits) address sizes?

### **hashed page table**

hashed page table

- Common in address spaces > 32 bit
- The virtual page number is hashed into a page table
- This page table contains a chain of elements hashing to the same location because the same hash function can have same value for different page no. .
- Each element contains (1) the virtual page number (2) the value of the mapped page frame (3) a pointer to the next element
- Virtual page numbers are compared in this chain searching for a match
- If a match is found, the corresponding physical frame is extracted

## **Q16:** What are the three components of a 32-bit ARM address?

- 1. Outer page**
- 2. Inner page**
- 3. Offset**

(Cache sits between main memory and CPU registers)

(Page number is used as an index into a page table)

(Page offset combined with the base address to define the physical memory address)

# Mass-Storage Structure

**Q1:** True or False? Magnetic disks provide the bulk of secondary storage for modern computer systems.

True

**Q2:** True or False? Solid state disks have the same characteristics as traditional hard disks.

True

**Q3:** What is the term for the smallest unit of transfer between a disk?

logical block

**Q4:** What are the two ways a computer can access disk storage?

1. **I/O ports** (host-attached storage)
2. **Distributed File Systems** (network-attached storage)

**Q5:** List the three general disk scheduling algorithms

1. **FCFS** (first come first serve)
2. **SSTF** (shortest seek time first)
3. **SCAN** (starts at one end and moves to the other, reversing back when it gets to each end)
- \*4. **C-SCAN** (Circular SCAN; instead of reversing it moves from one end to the other, then immediately proceeds back to the end it started scanning from)
- \*5. **LOOK** (Works like SCAN, but does not traverse the entire disk; LOOK only goes as far as the farthest request)

**Q6:** What disk scheduling algorithm is typically used with SSDs?

FCFS

**Q7:** What must be done to a disk before it can be used for storage?

**low-level formatting** (AKA physical formatting) - divided up into sectors that the disk controller can read and write.

**Q8:** Where does the Windows system place its boot code?

The MBR (master boot record) is placed, **in the first sector of a disk.**

**Q9:** True or False? Most modern systems swap entire processes to swap space.

**False;** Most modern systems **swap pages** to swap space, not entire processes.

**Q10:** What is the fundamental reason RAID is used?

**To address performance and reliability issues.**

(RAID (redundant array of independent disks) is a way of storing the same data in different places on multiple hard disks or solid-state drives (SSDs) to protect data in the case of a drive failure.)

**Q11:** What are the three possible outcomes when writing to disk?

- 1. Successful completion**
- 2. Partial failure** - failure after successfully writing some of the sectors
- 3. Total failure** - failure before disk write occurs.