Quiz 1

1. Which of the following is correct about batch systems on old mainframes?
    a. Multiple jobs can run in parallel on the same system.
    b. CPU is often fully utilized due to fast I/O speed.
    c. The system runs one user's job to completion at a time
    d. Multiple jobs can run concurrently on the same system.
    e. Users get to interact with the jobs using a keyboard and display time.

2. Multiprogramming addresses which of the following disadvantages with batch processing?
    a. Multiprogramming allows multiple users to share a machine, whereas batch systems cannot be shared by multiple users. →batch system also can be share by multiple users
    b. Multiprogramming means executing all the batch jobs of one user before all of another user's batch jobs. →it is batch system
    c. Multiprogramming runs one job to completion before starting the next job, whereas batch processing loads all jobs into memory at once. → kebalik
    d. Multiprogramming supports interactive user interface whereas batch processing does not. →interactive itu multitasking
    e. Multiprogramming keeps the processor utilized by performing computation and I/O in parallel.

3. Context switching is always done in which type of system?
    a. Mainframe system
    b. Multicore system
    c. Multiprogramming system
    d. Batch system
    e. Multitasking system

4. Which of the following is true about multiprocessor systems?
    a. An n-core multiprocessor automatically gives you n times the speedup.
    b. A symmetric multiprocessor can run different operating systems on different processors.
    c. A multicore multiprocessor can deliver the same performance as a single processor but at much lower power consumption.
    d. A multiprocessor system, regardless of symmetric or asymmetric, requires one processor to be master and the other processors to be slaves.
    e. In a loosely coupled system, the processors communicate with each other via shared memory. →supposed to be via bus

5. Which of the following is true about embedded and real-time systems?
    a. An OS is required for a system to make real-time
    b. A real-time system contains an electronic calendar and clock to track the current year, date, and time.
    c. An embedded system might not run an OS.
    d. An example of an embedded system is a laptop computer.
    e. All "fast" systems are real-time systems. guarantees.

6. Consider the assembly code in Intel 8051 assembly code

```
ORG 0020H
MOV 90H, #24H
END
```

   Which of the following is NOT a correct description of the effect of ORG 0020H ?
    a. It does not change the program counter.
    b. It is an assembler directive rather than an assembly instruction.
    c. ORG is the opcode and 0020H is the operand.
    d. It occupies no space in code memory.
    e. It places the MOV 90H, #24H instruction at address 0020H.

7. Which of the following 8051 assembly instruction is equivalent to MOV 90H, #24H?
    a. MOV 90H, #36 .
    b. MOV #144, 24H
    c. MOV #P1, #24H
    d. MOV 144, 36
    e. MOV #90H, 36

8. Which of the following is a correct description of END in the assembly source above?
    a. Nothing: it marks the end of the assembly source code, but the processor continues executing whatever is at that program memory location.
    b. It is an instruction for returning from the main() program.
    c. It is effectively an infinite loop to prevent the processor from executing anything after the program source code ends.
    d. It causes the processor to stop execution until the user presses the reset button.
    e. It causes the processor to jump to location 0000H and start running from the beginning.

9. Which of the following is correct about the sizes of various registers in 8051?
    a. PC is 8 bits
    b. RO. .R7 are 16 bits each
    c. PSW is 16 bits
    d. B is 32 bits
    e. DPTR is 16 bits

10. In the instruction MOV 23, #45, which of the following is correct about the operands?
    a. #45 is a direct operand.
    b. #45 is an immediate operand.
    c. 23 is a register operand.
    d. #45 is a register operand.
    e. 23 is an immediate operand.

11. Which of the following is most likely to be part of the kernel of a typical general-purpose OS?
    a. boot loader. Its closely related but not OS, it is to boot the OS
    b. compiler and linker.
    c. web server.
    d. web browser. → kinda consider as middleware (part of OS but not Core)
    e. device drivers.

12. Which of the following is not an example of a system program?
    a. Java interpreter.
    b. graphical user interface.
    c. text editor.
    d. program loader.
    e. command-line shell.

13. As a resource manager, what kinds of resources does an OS manage?
    a. command-line interface and graphical user interface commands.
    b. compiling and linking.
    c. processor caches and register files.
    d. CPU time, storage space, use of I/O devices.
    e. multimedia streaming and database.

14. Which of the following can a bootloader do?
    a. copy an application program image into RAM to spawn a new process.
    b. copy an OS image from a server into RAM and jump to it.
    c. make a system call to request an OS service.
    d. flush the processor cache in case of branch misprediction.
    e. respond to a database query.

15. Which of the following is true about I/O mechanisms?
    a. A processor performs an I/O by mapping one of its general-purpose registers such as RO..R7 to a status register or data register on the device controller.
    b. A processor performs an I/O by issuing an IRQ to the device controller.
    c. status register, data register, and buffer are part of the device controller.
    d. status register, data register, and buffer are part of the I/O device but are outside the device controller.
    e. CPU and device controller never talk to each other directly, but they always exchange data via shared memory,

16. How does a program for 8051 know when the UART has received a byte?
    a. TMOD register == 0x20.
    b. SBUF register contains a non-zero byte.
    c. RI flag == 0.
    d. SCON register == 0x50.
    e. RI flag==1.

17. Which of the following is true about polling vs. interrupt?
    a. polling is always slower than interrupt when responding to an I/O event.
    b. interrupts enable the processor to do useful work while waiting for an I/O event.
    c. polling and interrupt use the same ISR to handle the same I/O event.
    d. while waiting for an interrupt, the processor cannot do useful work or else it will miss the interrupt.
    e. polling sets up the processor to automatically call a subroutine when the polled flag becomes true.

18. Which of the following is correct about interrupts?
    a. An ISR is a subroutine called by the polling code to handle an I/O.
    b. An interrupt vector table is a data structure that contains the actual machine code for all of the ISRs of the processor.
    c. An interrupt vector is the address of anISR.
    d. An ISR is a conventional subroutine (returned using RET instruction on 8051) that just happens to be invoked by an interrupt.
    e. An interrupt vector is the data structure that contains the addresses of the ISRs.

19. How does a processor decide which ISR to execute when multiple I/O devices trigger their IRQ on different interrupt lines at the same time?
    a. Neither ISR will be executed due to the IRQ conflict.
    b. the higher-priority interrupt will be taken while the lower-priority interrupt has to wait.
    c. the device whose flag is polled first is executed.
    d. the ISR for the input device will always be executed before the ISR for the output device.
    e. the processor executes the ISR for both I/O devices in parallel.

20. How does an OS protect the CPU time as a resource by preventing a user program from hogging the CPU?
    a. The OS sets up a base+limit register to track the number of times the user executes a loop and terminates the program if it exceeds the bound.
    b. The OS executes a privileged instruction such as ERET on MIPS to force a mode change from user mode to kernel mode.
    c. The OS executes privileged instructions such as syscall in MIPS to force a mode change from user mode to kernel mode.
    d. The OS forces the user program to make a system call.
    e. The OS uses timer interrupt to gain control.

21. (preview) What does a loader do?
    a. It reads the .hex file into EdSim51's instruction memory so that it can be shown as assembly code and run.
    b. It mounts a disk drive so that the OS can read and write the data.
    c. It reads the program image from disk to memory and starts the process.
    d. It loads a cache line from main memory into the processor's cache.
    e. It loads an array of words from a cache line into consecutively numbered registers in the processor so it would not incur a cache miss.

Quiz 2

1. Which of the following is true about two major approaches to a shell ?
    a. A shell that understands all commands runs in user mode whereas a shell that invokes executable files runs in kernel mode
    b. A shell that understands all commands runs in kernel mode whereas a shell that invokes executable files runs in user mode.
    c. A shell that invokes executable files requires modification if you need to change commands.
    d. A shell that understands all commands is more efficient per command
    e. A shell that understands all commands is easily expandable but is heavier weight per command

2. What is the purpose of a system call
    a. A synonym for an interrupt service routine
    b. For the OS to make a function call in kernel mode
    c. For a user program to request an OS service to be performed in kernel mode.
    d. For a shell to interpret a command from the user.
    e. For the OS to notify the user program of a system event by making callback in user mode

3. How does a system call instruction indicate what service to request ?
    a. It pushes the target address of the system-call routine on the stack
    b. It writes the system call number to disk
    c. It passess the system call number in a register.
    d. It takes the target address of the system-call routine.
    e. It passess the target address of the system-call routine in a register.

4. What is the difference between system calls and API?
    a. An API runs entirely in kernel mode, whereas system call runs entirely in user mode.
    b. A system call traps a kernel and is commonly written in assembly code.
    c. A system call may be done entirely in user mode without necessarily trapping to kernel
    d. A system call makes a transition from user mode to kernel mode, whereas an APi makes a transition from kernel mode to user mode.
    e. An API makes at least one system call

5. What is the principle of separation between policy and mechanism
    a. To maximize safety if policy decisions are to be changed later.
    b. To allow optimization of mechanism as long as the policy is unchanged
    c. To write code in a way that can run efficiently on both uniprocessors and multiprocessors without modification
    d. To allow maximum flexibility if policy decisions are to be changed later
    e. To maximize performance for all combinations of policy and mechanisms

6. Which of the following is true about monolithic kernels ?
    a. It is simple by supporting single-tasking only but can not support multitasking.
    b. It offers no protection because it runs both the kernel and user code in the same address space.
    c. It is difficult to define the boundary between kernel mode and user mode.
    d. It is easy to debug and maintain but higher overhead than microkernel
    e. It is lower overhead compared to microkernels but is hard to scale unless organized in a modular structure.

7. Which of the following functions of a microkernel is done (i.e., executed) in kernel mode or user mode ?
    a. CPU Scheduling is done in user mode.
    b. Device driver is done in kernel mode.
    c. Interprocess communication (IPC) is done in kernel mode.
    d. Memory management is done in user mode.
    e. File system is done in kernel mode.

8. How can a microkernel run an OS service in user mode while also protecting the rest of the system ?
    a. Each service actually runs in a special user mode that can access privileged instructions but is limited to only a small segment of memory.
    b. Each service runs in its own address space and cannot affect the rest of the system
    c. Each service may run in user mode but hardware interrupts still run in kernel mode.
    d. Each service runs in the same address space as the user making the system call and can only trash that user's data.
    e. Each service is loaded on demand as a loadable kernel module and therefore there is nothing to trash when the service is not running.

9. Why does SDCC require that you declare an ISR with __ interrupt(4) as in void Serial_ISR(void)__interrupt(4)?
    a. It associates the Serial_ISR routine with the interrupt vector for UART and uses RETI for return.
    b. It sets the address of the JSR for UART to 0x4.
    c. It allocates 4 bytes in the internal RAM for use by the ISR for UART.
    d. It automatically saves four registers R0 .. R3 so they can be used by the ISR without overhead.
    e. It allocates 4 bytes in the program memo,y for the ISR for UART.

10. If you need to call the main() function from inlined 8051 assembly in SDCC, what do you have to write?
    a. lcall __ main__
    b. lcall _main
    c. lcall #main
    d. lcall main
    e. lcall __ main

11. What is the difference between a program, process, job, and task?
    a. a process is the preferred term to a program executable, whereas a task is the preferred term to a running job.
    b. a program is the static executable, while a process is an instance of a program in execution.
    c. a process is an instance of a task running in kernel mode. whereas a job is a program running in user mode.
    d. a process is a program running in kernel mode, whereas a job is a task running in user mode.
    e. a task is not yet scheduled, whereas a job has been scheduled.

12. What are the sections of memory of a process?
    a. The heap section contains priority queue data structures commonly used for schedulers.
    b. The code section contains encrypted data such as passwords.
    c. The stack section contains the global variables and dynamically allocated variables.
    d. The stack section contains auto-local variables of functions, parameters passed to a function call, and return address.
    e. The text section contains plain text strings encoded in ASCII or Unicode.

13. Which of the following is NOT a valid description of process state?
    a. terminated state means the process has finished execution and its space can be reclaimed.
    b. waiting state means the process can be executed but it is waiting to be dispatched in a queue.
    c. ready state means the process is in memory and can be assigned to a processor but is not currently running.
    d. running means the process is being executed by the processor.
    e. new state means the process is being created by the OS.

14. What can trigger a context switch?
    a. when allocating or deallocating a process control block (PCB).
    b. when calling enqueue() or dequeue() on a process scheduling queue.
    c. when making a subroutine call or return.
    d. an interrupt or a system call.
    e. when saving (pushing) or restoring (popping) registers to or from a stack.

15. In Unix-like systems, which of the following is NOT a correct description of what fork() does?
    a. It returns the child's PID lo the parent and returns 0 to the child.
    b. Modern implementations of fork() use copy-on-write to be more efficient.
    c. It creates a child process by cloning the parent process.
    d. The child process and the present process execute concurrently after the fork().
    e. It spawns a new process by copying the program code from the executable file specified by the parameter.

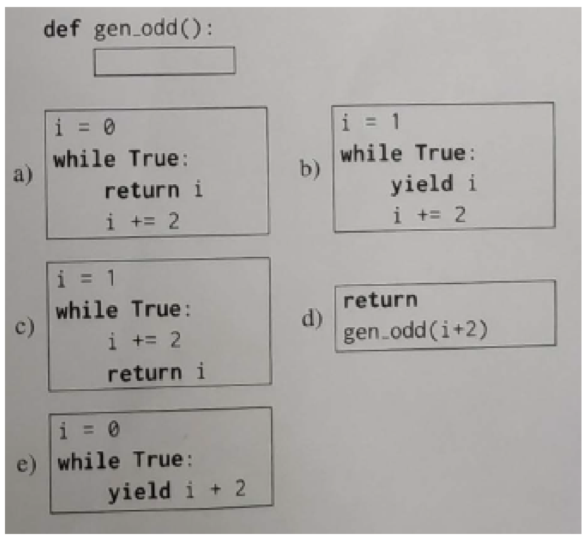
16. Which of the following is true about process termination in a Unix-like system?
    a. In cascaded termination, killing a child process also terminates its parent process .
    b. when a child process terminates, it should call wait() on its parent process to ensure its PID can be recycled.
    c. a parent process can terminate a child process by calling exit() with the child's PID as argument.
    d. a process implicitly calls exit() upon return from main() .
    e. A zombie process is a child process that is alive when its parent process has died.

17. Which of the following is true about shared memory vs. message passing for interprocess communication?
    a. shared memory requires rendezvous synchronization while message passing uses a shared FIFO buffer for blocking send() and receive() .
    b. shared memory and message passing both call send() and receive() except shared memory calls are non blocking while message passing calls are blocking.
    c. shared memory is faster because the OS is involved only during setup but synchronization is trickier.
    d. message passing is more efficient than shared memory for transferring large amounts of data.
    e. shared memory is for two-way (duplex) communication whereas message passing is for one-way (simplex) communication.

18. Which of the following is true about the sender and receiver in bounded-buffer communication?
    a. In the unbounded buffer, the receiver never blocks but the sender blocks only if the buffer is empty .
    b. in an unbounded buffer, the sender never blocks, and the receiver blocks only if the buffer is empty.
    c. in zero buffer, the sender must always arrive earlier at the rendezvous point before the receiver does.
    d. in zero buffer, the process that arrives at rendezvous point earlier does not block, while the process that arrives later blocks.
    e. in a bounded buffer, the sender blocks if the buffer is empty, and the receiver blocks if the buffer is full.

19. What are some advantages of generators in Python over interprocess communication between a producer process and a consumer process?
    a. The producer as a generator and consumer as the main program run in different address spaces and are protected from each other.
    b. It is easier to write and understand code for the producer and consumer.
    c. The producer as a generator uses the same stack as the consumer and therefore uses less space.
    d. The producer as a generator passes a message to the consumer for better fairness of scheduling.
    e. The producer as a generator and the consumer can run in parallel on multiple processors.

20. Which of the following correctly implements a generator of odd numbers in Python?



```
def gen_odd():
    [          ]
```

a)
```
i = 0
while True:
    return i
    i += 2
```

b)
```
i = 1
while True:
    yield i
    i += 2
```

c)
```
i = 1
while True:
    i += 2
    return i
```

d)
```
return
gen_odd(i+2)
```

e)
```
i = 0
while True:
    yield i + 2
```

i = 1
while True:
Yield i
i += 2

21. (Preview) What is the difference between a thread and a process?
    a. Threads are for cooperative multitasking, whereas processes can be preempted by the kernel.
    b. Different threads share the same runtime stack, whereas different processes have their own runtime stacks.
    c. A context switch between threads is lighter weight than between processes.
    d. A process can spawn other processes, but a thread cannot spawn other threads.
    e. Different threads run concurrently, while different processes run in parallel.

Quiz 3

1. Which of the following is a correct description of processes vs thread ?
    a. A thread runs until completion whereas a process can be preempted
    b. A thread runs in user mode while a process runs in privileged mode.
    c. Each thread has its own address space, but processes of a given thread share the same address space.
    d. Each process has its own address space, but threads of a given process share the same address space.
    e. A process is an instance of a single thread program in execution

2. Why is it a good idea to structure a server by spawning a thread to serve a new request compared to spawning a new process?
    a. better performance because threads can share memory with the client but processes cannot.
    b. more secure because each thread runs in its own address space.
    c. much lower overhead due to sharing of code and resources within the same server process.
    d. better performance because threads can use message passing IPC with the client but processes can use only shared-memory IPC.
    e. much lower latency because different threads share the same stack.

3. Which of the following is true about RPC?
    a. An RPC function establishes a shared-memory region with the RPC server for passing parameters and return values.
    b. An RPC function requires the client and server processes to run the same instruction-set architecture for the data types and pointers to be compatible.
    c. A stub function marshals the parameters, requests the remote server to do the actual work, and demarshals the reply message as return value.
    d. An RPC function is one where the caller opens a network connection to the server and passes the socket to the stub function to make the call.
    e. An RPC function spawns a server process with fork() before sending that process image to a remote machine for execution.

4. Which of the following is true about concurrency and parallelism?
    a. Concurrency is restricted to kernel mode while parallelism is in user mode.
    b. Concurrency means possibility of running in parallel, whereas parallel means simultaneous execution.
    c. On a uniprocessor, threads run in parallel within a process but different processes run concurrently.
    d. On a multiprocessor, different processors run concurrently but the processes running on a given processor are in parallel.
    e. Concurrency and parallelism are used interchangeably, although concurrency is the preferred term.

5. Which of the following is true about kernel threads?
    a. threads that are spawned by the user but are currently running in kernel mode.
    b. an alternative term for processes.
    c. a user thread that is allowed to share the kernel's memory.
    d. threads that are managed by the kernel but may be assigned to user threads.
    e. threads that always run in kernel mode.

6. Which of the following is true about converting recursive calls to Quicksort into multiple threads on a multi-core or multiprocessor ?
    a. It runs at exactly the same speed as the single-threaded version because Quicksort contains no concurrency
    b. It runs at exactly the same speed a the single-threaded version because Quicksort contains no parallelism
    c. It runs faster only if the number of threads does not exceed the number of cores.
    d. It always runs faster than the serial version due to parallel execution.
    e. It may run slower than the serial version even on a multi-core due to overhead of thread management.

7. Which of the following is correct about scheduler activation ?
    a. The kernel moves the scheduler process from the wait queue to the ready queue.
    b. the user-level thread manager spawns a new thread to schedule the other thread .
    c. the kernel allocates a kernel thread to a user thread to prevent it from blocking the entire process.
    d. The user-level thread manager invokes the scheduler to compute a schedule for all ready but not running threads.
    e. the user thread makes a system call to the kernel to activate the process scheduler.

8. What is the relationship between a kernel thread and a hardware thread?
    a. "kernel thread" is the preferred term for "hardware thread" but they mean the same thing.
    b. one or more kernel threads may run on a given hardware thread.
    c. kernel threads require at least two hardware threads to work.
    d. kernel thread and hardware thread must map one-to one.
    e. one kernel thread may run simultaneously on multiple hardware threads.

9. What is the difference between join() and detach() in the context of threads?
    a. join() is called by the main thread to free up a thread's resource upon its termination, whereas detach() marks the thread to be automatically reclaimed on termination.
    b. join() is called by the child thread when it wants to yield control to the main thread, whereas detach() kills a child thread.
    c. The main thread calls detach() on a thread to run it and later calls join() to free up its resource.
    d. The main thread calls detach() to clone itself and join() to wait for a child thread to finish.
    e. join() is called by different threads for rendezvous synchronization, whereas detach() unblocks a thread.

10. What does it mean to cancel(t) asynchronously, where t is a thread?
    a. it reclaims the resources of a blocked thread t.
    b. it asks thread t to terminate itself.
    c. it moves thread t from ready-queue to wait-queue.
    d. It forces thread t to terminate.
    e. reclaims the resources of a terminated thread t.

11. If you have a Python statement g. send(20) where g is an instance of a generator, how does the generator receive the value and put it in a local variable c?
    a. c = g.recv()
    b. c = yield c
    c. yield c
    d. g.recv(c)
    e. c = next(g)

12. What is the type of parallelism supported in OpenMP?
    a. task parallelism.
    b. control parallelism.
    c. data parallelism.
    d. fork-join parallelism.
    e. pipeline parallelism.

13. How does the RET instruction in 8051 know where to return to?
    a. The return address is in the PSW register.
    b. The return address is found in the top two bytes of the stack.
    c. The return address is in the SP register.
    d. the return address is found as a two-byte immediate operand of the RET instruction.
    e. The return address is in DPTR, which is concatenated from DPH and DPL.

14. Which of the following sequences of 8051 instructions allows you to transfer control to (i.e .. effectively jump to) the (code) address contained in DPTR?
    a. MOV DPTR, PC
    b. MOV PC, DPTR
    c. LJMP DPTR
    d. PUSH DPL
       PUSH DPH
       RET
    e. LCALL DPTR

15. How does SDCC pass a single-byte parameter?
    a. In DPTR
    b. In R0
    c. In DPH
    d. Pushing the byte on the stack
    e. In DPL

16. If you define a function named Bootstrap in SDCC, what is the corresponding label in assembly?
    a. ?Bootstrap
    b. __Bootstrap
    c. Bootstrap
    d. _Bootstrap
    e. $Bootstrap$

17. For the threads project on Edsim51 , how are the stacks for the four threads allocated in the on-chip RAM? The H suffix below indicates hexadecimal base.
    a. 20H-27H, 28H-2FH, 30H-37H, 38H-3FH
    b. 00H-0FH, 10H-1FH, 20H-2FH, 30H-3FH
    c. 40H-4FH, 50H-5FH, 60F-6FH, 70F-7FH
    d. 00H-07H, 08H-0FH, 10H-17H, 18H-1 FH
    e. 80H-8FH, 90H-9FH, A0H-AFH, B0H-BFH

18. What is the power-on initial value of the stack pointer (SP) on 8051?
    a. 7FH
    b. 07H
    c. 00H
    d. 80H
    e. FFH

19. In SDC, how do you make the stack pointer point to address 46H, assuming you have #include <8051.h> ?
    a. SP << 0X46;
    b. SP = 0X46;
    c. You can't: a stack pointer cannot be modified.
    d. sp = 0X46;
    e. _SP = 0X46;

20. In 8051, what did PSW stand for ?
    a. Preemptive scheduler word.
    b. Program status word.
    c. Protected status word.
    d. Processor stack word.
    e. Process state word.

21. (preview) Which of the following is correct about scheduling policies?
    a. SJF is shortest-job-first scheduling.
    b. RR is repeat-rate scheduling.
    c. FCFS is fastest-completed-first scheduling.
    d. EDF is exponential-deadline-first scheduling.
    e. RM is rotated-multiprocessor scheduling.

Quiz 4

1. Which of the following is not one of the four cases when *preemptive* CPU schedulers can take control?
   a. Upon a timer interrupt
   b. Upon I/O completion
   c. Upon making a system call
   d. Upon process termination
   e. Upon starting an I/O

2. What is the conceptual difference between a dispatcher and a scheduler?
   a. The two terms are synonymous, but scheduler is preferred
   b. The two terms are synonymous, but the dispatcher is preferred
   c. The scheduler switches to the process chosen by the dispatcher
   d. The dispatcher switches to the process chosen by the scheduler
   e. A scheduler schedules processes, whereas a dispatcher schedules threads

3. Which of the following correctly defines one of the scheduling criteria?
   a. CPU utilization is the amount of time a process spends in running state
   b. Waiting time is the amount of time that a process spends in the wait queue
   c. Response time is the amount of time from job submission to completion
   d. Throughput is the amount of time a process spends in the ready queue
   e. Turnaround time is the amount of time from job to submission to completion

4. Which of the following is true about shortest job first (SJF) scheduling?
   a. It assumes a nonpreemptive kernel
   b. It assumes a preemptive kernel
   c. It is the default policy in linux
   d. It selects the job with the shortest next cpu burst
   e. It selects the job with the shortest total length

5. Which of the following is true about first come first serve (FCFS) scheduling?
   a. It minimizes average waiting time
   b. It is a theoretical optimal algorithm but cannot be implemented in practice
   c. Longer jobs can suffer from starvation
   d. Shorter jobs can suffer from the convoy effect
   e. It minimizes total waiting time

6. Which of the following is true about round-robin (RR) scheduling?
   a. For long time quantum, RR may suffer from starvation
   b. For short time quantum, RR becomes similar to SJF
   c. For short time quantum, RR becomes similar to FCFS
   d. For long time quantum, RR becomes similar to FCFS
   e. For long time quantum, RR becomes similar to SJF

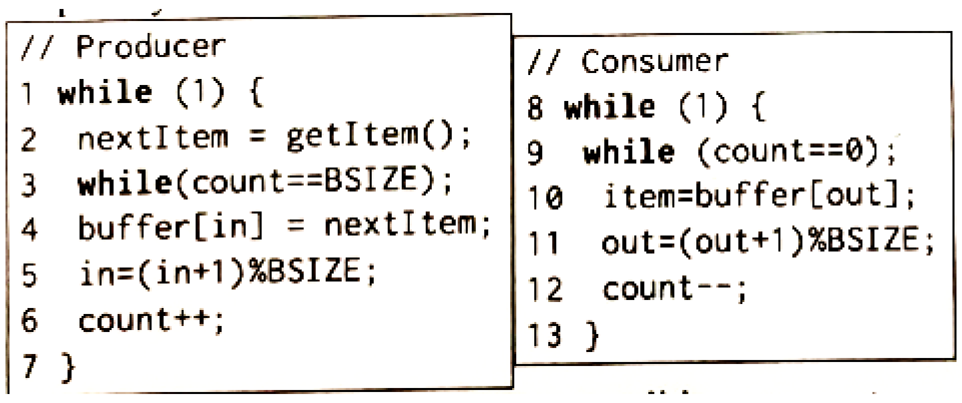
7. What is one way of preventing starvation?
   a. wait()
   b. Preemption
   c. Aging
   d. Context switching
   e. fork()

8. Which of the following is true about multiprocessor scheduling?
   a. Asymmetric multiprocessing means multiple slave processors execute the schedule computed by one single master processor
   b. Symmetric multiprocessing means all processors spawn the same set of processes but may schedule them in its own way
   c. Symmetric multiprocessing means the same scheduling algorithm runs on all processors
   d. Asymmetric multiprocessor means each processor runs its own different scheduling algorithm
   e. Symmetric multiprocessing means all processors execute the same schedule computed by a single master processor

9. Which of the following is true about rate monotonic scheduling, assuming no resource sharing?
   a. The priority depends on only the period but not the execution time of the task
   b. The necessary condition for schedulability is the CPU utilization $\leq \ln 2$
   c. The priority depends on the execution time of the task but not its period
   d. RM scheduling is the same as SJF if each task is spawned immediately after its period
   e. The priority of each task is updated on every context switch

10. Which of the following is true about earliest deadline first (EDF) vs rate monotonous (RM) scheduling?
    a. RM is for aperiodic tasks only, whereas EDF is for periodic tasks only
    b. RM is nonpreemptive, whereas EDF is preemptive
    c. RM is preemptive, whereas EDF is nonpreemptive
    d. RM tasks have static priority whereas EDF tasks have dynamic priority
    e. RM is for periodic tasks only, whereas EDF is for aperiodic tasks only

11. Consider the following producer consumer code, where the producer and consumer run concurrently and preemptively:



```
// Producer
1 while (1) {
2  nextItem = getItem();
3  while(count==BSIZE);
4  buffer[in] = nextItem;
5  in=(in+1)%BSIZE;
6  count++;
7 }
```

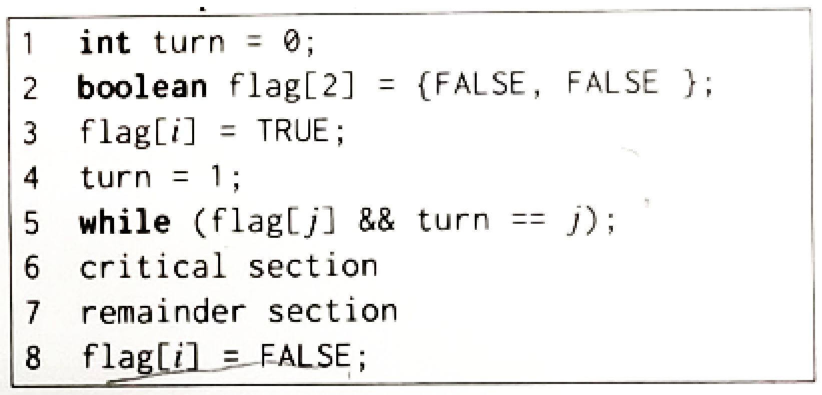
```
// Consumer
8 while (1) {
9  while (count==0);
10  item=buffer[out];
11  out=(out+1)%BSIZE;
12  count--;
13 }
```

    Which variable could have a race condition on most RISC like processors?
    a. Out
    b. Count
    c. nextItem
    d. Item
    e. in

12. Continuing with the previous question, where can you put critical sections to eliminate the race condition?
    a. Around line 4 and another around line 10
    b. Around line 6 and another around line 12
    c. Around line 5 and another around line 11
    d. Around line 2 and another around line 10
    e. Around line 3 and another around line 9

13. Which of the following is NOT a requirement of a critical section (CS)?
    a. When no process is in its CS, a process wishing to enter its CS must not block indefinitely
    b. There exist a limit on the number of times that other processes are allowed to enter their CS after a process has made its request to enter its CS
    c. A bound must exist on the number of times that other processes are allowed to enter their CS after the process has made a request to enter its CS
    d. After a process has requested to enter its CS, other processes are allowed to enter their CS for a finite number of times
    e. At most one process is executing in its CS at any same time

14. Which of the following is true about Peterson;s solutions to the critical section problem? An excerpt is as follows for process Pi:



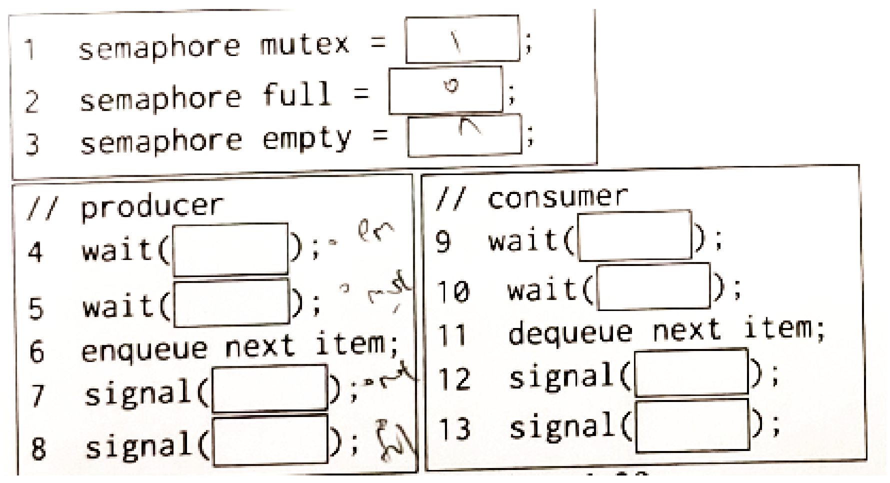
```
int turn = 0;
boolean flag[2] = {FALSE, FALSE };
flag[i] = TRUE;
turn = 1;
while (flag[j] && turn == j);
critical section
remainder section
flag[i] = FALSE;
```

    a. Peretson's solution assumes non preemptive scheduler or else it will not work
    b. Bounded wait is achieved because while loop's condition for i will always have an AND condition broken by j if not already
    c. Peterson's solution works on multicore architectures as long as you limit the number of processes to two
    d. Progress is achieved because after process Pi finishes its critical section, it always sets its own flag[i] = FALSE and turn is already == j
    e. Mutual exclusion is achieved by the variable turn when flag[i] == flag[j] == TRUE

15. Which of the following is true about atomic TestAndSet, which works on a boolean variable named lock?
    a. TestAndSet(lock) must evaluate to FALSe before the process can exit its critical section
    b. TestAndSet(lock) returns TRUE if the process successfully acquires the lock and sets lock to FALSE as side effect
    c. To lock, simply set lock = TRUE;
    d. TestandSet(lock) must evaluate to TRUE before the process can enter its critical section
    e. To unlock, simply set lock = FALSE;

16. Which of the following is true about wait(S) and signal(S), where S is a semaphore?
    a. wait(S) never blocks, but signals(S) always blocks
    b. Both wait(S) and signal(S) may block
    c. wait(S) always blocks, but signal(S) never blocks
    d. wait(S) may block, but signal(S) never blocks
    e. wait(S) never blocks, but signal(S) may block

17. Given the classical bounded n-buffer problem written using semaphores,



```
1  semaphore mutex = [ \ ];
2  semaphore full  = [ 0 ];
3  semaphore empty = [ n ];

// producer                     // consumer
4  wait(        );              9  wait(        );
5  wait(        );              10 wait(        );
6  enqueue next item;           11 dequeue next item;
7  signal(        );            12 signal(        );
8  signal(        );            13 signal(        );
```

What are the initial values for lines 1-3?
    a. 1, 0, n
    b. 1, n, 0
    c. 0, 1, n
    d. 1, n, n
    e. 0, 0, 0

18. What values should be on lines 4, 5, 7, 8?
    a. Empty, mutex, mutex, empty
    b. Empty, mutex, mutex, full
    c. Empty, mutex, full, mutex
    d. Mutex, empty, full, mutex
    e. Full, mutex, mutex, empty

19. In the readers-writers classical synchronization problem, two semaphores named rw_mutex and mutex are declared. What are their purposes?
    a. Mutex allows at most one reader to be reading the shared buffer
    b. Rw_mutex allows at most one writer or unlimited number of readers to be accessing the shared buffer
    c. Rw_mutex allows either all writers or all readers to access the shared buffer
    d. Rw_mutex allows at most one writer to block out all other processes or the first reader to block all writers
    e. Mutex allows at most one writer or one reader to be accessing the shared buffer

20. Which of the following is a correct usage of barrier synchronization implemented using a semaphore b, where S1 of process P1 should be executed before S2, of process P2?
    a. Process P1 should be written as S1; wait(b);
    b. B should be initialized to 1
    c. Process P2 should be written as S2; signal(b);
    d. Process P2 should be written as wait(b); S2;
    e. Process P1 should be written as signal(b); S1;

21. (preview) What is the difference between a deadlock and a livelock?
    a. All processes in a deadlock are blocked, whereas all processes in a livelock can all be dispatched.
    b. A deadlock causes a process to die, but a livelock keeps it running.
    c. A deadlock crashes the machine, but a livelock does not.
    d. Processes in a deadlock fail to acquire a mutex lock, whereas processes in a livelock fail to acquire a counting semaphore.
    e. All processes in a deadlock are in the ready queue, whereas all processes in a livelock are in the wait queue.

MIDTERM

1. Multiprogramming addresses which of the following disadvantages with batch processing?
a)Multiprogramming keeps the processor utilized by performing computation and I/O in parallel.
b) Multiprogramming runs one job to completion before starting the next job, whereas batch processing loads all jobs into memory at once.
c) Multiprogramming supports interactive user interface whereas batch processing does not.
d) Multiprogramming allows multiple users to share a machine, whereas batch systems cannot be shared by multiple users.
e) Multiprogramming means executing all the batch jobs of one user before all of another user's batch jobs.

2. Which of the following is true about multiprocessor systems?
a) An n-core multiprocessor automatically gives you n times the speedup.
b) A multi-core multiprocessor can deliver the same performance as a single processor but at much lower power consumption.
c) A multiprocessor system, regardless of symmetric or asymmetric, requires one processor to be master and the other processors to be slaves.
d) A symmetric multiprocessor can run different operating systems on different processors.
e) In a loosely coupled system, the processors communicate with each other via shared memory.

3. Which of the following is true about embedded and real-time systems?
a) An embedded system might not run an OS.
b) An example of an embedded system is a laptop computer.
c) A real-time system contains an electronic calendar and clock to track the current year, date, and time.
d) An OS is required for a system to make real-time guarantees.
e) All "fast" systems are real-time systems.

4. Which of the following is NOT a correct description of the effect of ORG 0020H
a) It places the MOV 98H, #24H instruction at address 0020H.
b) It is an assembler directive rather than an assembly instruction.
c) It does not change the program counter.
d) It occupies no space in code memory.
e) ORG is the opcode and 0020H is the operand

5. Which of the following is correct about the sizes of various registers in 8051?
a) DPTR is 16 bits
b) B is 32 bits
c) PSW is 16 bits
d) Re..R7 are 16 bits each
(e) PC is 8 bits

6. Which of the following is most likely to be part of the kernel of a typical general-purpose OS?
a) web server
b) boot loader.
c) web browser.
d) device drivers.
e) compiler and linker.

7. Which of the following can a bootloader do?
a) copy an OS image from a server into RAM and jump to it.
b) respond to a database query. X
c) make a system call to request an OS service.
d) flush the processor cache in case of branch mispre- diction.
e) copy an application program image into RAM to spawn a new process.

8. Which of the following is true about I/O mechanisms?
a) A processor performs an I/O by mapping one of its general-purpose registers such as RO..R7 to a status register or data register on the device controller.
b) status register, data register, and buffer are part of the device controller.
c) CPU and device controller never talk to each other directly, but they always exchange data via shared memory.
d) status register, data register, and buffer are part of the I/O device but are outside the device controller.
e) A processor performs an I/O by issuing an IRQ to the device controller.

9. Which of the following is correct about interrupts?
a) an ISR is a subroutine called by the polling code to handle an I/O.
b) an interrupt vector is the data structure that contains the addresses of the ISRS.
c) an interrupt vector is the address of an ISR.
d) an interrupt vector table is a data structure that contains the actual machine code for all of the ISRS of the processor.
e) an ISR is a conventional subroutine (returned using RET instruction on 8051) that just happens to be invoked by an interrupt,

10. How does an OS protect the CPU time as a resource by preventing a user program from hogging CPU?
a) The OS executes a privileged instruction such as syscall in MIPS to force a mode change from user mode to kernel mode.
b) The OS sets up base+limit register to track the num- ber of times the user executes a loop and terminates the program if it exceeds the bound.
c) The OS uses timer interrupt to gain control.
d) The OS executes a privileged instruction such as ERET on MIPS to force a mode change from user mode to kernel mode.
e) The OS forces the user program to make a system call.

11. Which of the following is true about two major ap- proaches to a shell?
a) A shell that understands all commands is more efficient per command.
b) A shell that understands all commands runs in kernel mode whereas a shell that invokes executable files runs in user mode.
c) A shell that understands all commands is easily expandable but is heavier weight per command.
d) A shell that invokes executable files requires modifi- cation if you need to change commands.
e) A shell that understands all commands runs in user mode whereas a shell that invokes executable files runs in kernel mode.

12. How does a system call instruction indicate what ser- vice to request?
3) It passes the target address of the system-call routine in a register.
b) It pushes the target address of the system-call routine on the stack.
c) It takes the target address of the system-call routine.
d) It writes the system call number to disk.
e) It passes the system call number in a register.

13. What is the principle of separation between policy and mechanism?
a) to write code in a way that can run efficiently on both uniprocessors and multiprocessors without modifica- tion
b) to maximize safety if policy decisions are to be changed later.
c) to maximize performance for all combinations of pol- icy and mechanisms.
d) to allow maximum flexibility if policy decisions are to be changed later.
e) to allow optimization of the mechanism as long as the policy is unchanged.

14. Which of the following is true about monolithic kernels?
a) it is simple by supporting single-tasking only but cannot support multitasking.
b) it is easy to debug and maintain but is higher overhead than microkernel.
c) it offers no protection because it runs both the kernel and user code in the same address space.
d) it is difficult to define the boundary between kernel mode and user mode.
e) it is lower overhead compared to microkernel but is hard to scale unless organized in a modular structure.

15. How can a microkernel run an OS service in user mode while also protecting the rest of the system?
a) each service actually runs in a special user mode that can access privileged instructions but is limited to only a small segment of memory.
b) each service runs in the same address space as the user making the system call and can only trash that user's data.
c) each service is loaded on demand as a loadable ker- nel module and therefore there is nothing to trash when the service is not running.
d) each service may run in user mode but hardware interrupts still run in kernel mode.
e) each service runs in its own address space and cannot affect the rest of the system.

16. If you need to call the main() function from inlined 8051 assembly in SDCC, what do you have to write?
a) lcall main
b) lcall _main
c) lcall __main___
d) lcall #main
e) lcall__main

17. In Unix-like systems, which of the following is NOT a correct description of what fork() does?
a) It returns the child's PID to the parent and returns 0 to the child.
b) Modern implementations of fork() use copy-on-write to be more efficient.
c) It spawns a new process by copying the program code from the executable file specified by the parameter.
d) The child process and the parent process execute concurrently after the fork().
e) It creates a child process by cloning the parent pro- cess.

18. Which of the following is true about process termi- nation in a Unix-like system?
a) in cascaded termination, killing a child process also terminates its parent process.
b) a zombie process is a child process that is alive when its parent process has died.
c) when a child process terminates, it should call wait() on its parent process to ensure its PID can be recycled.
d) a process implicitly calls exit() upon return from main().
e) a parent process can terminate a child process by calling exit() with the child's PID as argument.

19. Which of the following is true about shared memory vs. message passing for interprocess communi- cation?
a) message passing is more efficient than shared memory for transferring large amounts of data.
b) shared memory is faster because the OS is involved only during setup but synchronization is trickier.
c) shared memory is for two-way (duplex) communica- tion whereas message passing is for one-way (sim- plex) communication.
d) shared memory requires rendezvous synchronization while message passing uses a shared FIFO buffer for blocking send() and receive().
e) shared memory and message passing both call send() and receive() except shared memory calls are non- blocking while message passing calls are blocking.

20. What are some advantages of generators in Python over interprocess communication between a producer process and a consumer process?
a) The producer as a generator and the consumer can run in parallel on multiple processors.
b) The producer as a generator and consumer as the main program run in different address spaces and are protected from each other.
c) It is easier to write and understand code for the producer and consumer.
d) The producer as a generator passes a message to the consumer for better fairness of scheduling.
e) The producer as a generator uses the same stack as the consumer and therefore uses less space.

21. Which of the following is true about RPC?
a) An RPC function establishes a shared-memory region with the RPC server for passing parameters and return values.
b) A stub function marshals the parameters, requests the remote server to do the actual work, and demarshals the reply message as return value.
c) An RPC function spawns a server process with fork() before sending that process image to a remote ma- chine for execution.
d) An RPC function is one where the caller opens a net- work connection to the server and passes the socket to the stub function to make the call.
e) An RPC function requires the client and server pro-
cesses to run the same instruction-set architecture for the data types and pointers to be compatible.

22. Which of the following is true about converting recursive calls to Quicksort into multiple threads on a multicore or multiprocessor?
a) It runs faster only if the number of threads does not exceed the number of cores.
b) It may run slower than the serial version even on a multi-core due to overhead of thread management.
c) It runs at exactly the same speed as the single-threaded version because Quicksort contains no parallelism.
d) It runs at exactly the same speed as the single-threaded version because Quicksort contains no concurrency.
e) It always runs faster than the serial version due to parallel execution.

23. Which of the following is correct about scheduler activation?
a) the kernel allocates a kernel thread to a user thread to prevent it from blocking the entire process.
b) the user-level thread manager spawns a new thread to schedule the other threads.
c) the user thread makes a system call to the kernel to activate the process scheduler.
d) the user-level thread manager invokes the scheduler to compute a schedule for all ready but not running threads.
e) the kernel moves the scheduler process from wait queue to ready queue.

24. What is the relationship between a kernel thread and a hardware thread?
a) one kernel thread may run simultaneously on multiple hardware threads.
b) "kernel thread" is the preferred term for "hardware thread" but they mean the same thing.
c) kernel thread and hardware thread must map one-to- one.
d) kernel threads require at least two hardware threads to work.
e) one or more kernel threads may run on a given hardware thread.

25. If you have a Python statement g. send (20) where g is an instance of a generator, how does the generator receive the value and put it in a local variable c?
a) c = g.recv()
b) c = yield c
c) c = next (g)
d) yield c
e) g.recv(c)

26. What is the type of parallelism supported in OpenMP?
a) task parallelism.
b) data parallelism.
c) fork-join parallelism.
d) control parallelism.
e) pipeline parallelism.

27. How does the RET instruction in 8051 know where to return to?
a) the return address is found in the top two bytes of the stack.
b) the return address is in DPTR, which is concatenated from DPH and DPL.
c) the return address is in the PSW register.
d) the return address is found as a two-byte immediate operand of the RET instruction.
e) the return address is in the SP register.

28. Which of the following sequences of 8051 instructions allows you to transfer control to (ie., effectively jump to) the (code) address contained in DPTR?
a) LCALL DPTR
b) MOV DPTR, PC
c) MOV PC, DPTR
d) PUSH DPL
PUSH DPH
RET
e) LJMP DPTR

29. In SDCC, how do you make the stack pointer point to address 46H, assuming you have #include <8051.h>?
a) _SP = 0x46;
b) sp = 0x46;
c) You can't: a stack pointer cannot be modified.
d) SP = 0x46;
e) SP <<<<0x46;

30. In 8051, what does PSW stand for?
a) processor stack word.
b) program status word.
c) process state words.
d) process scheduler word.
e) protected stack word.

31. Which of the following is not one of the four cases when a preemptive CPU scheduler can take control?
a) on I/O completion
b) on a timer interrupt
c) on starting an I/O
d) on making a system call
e) on process termination

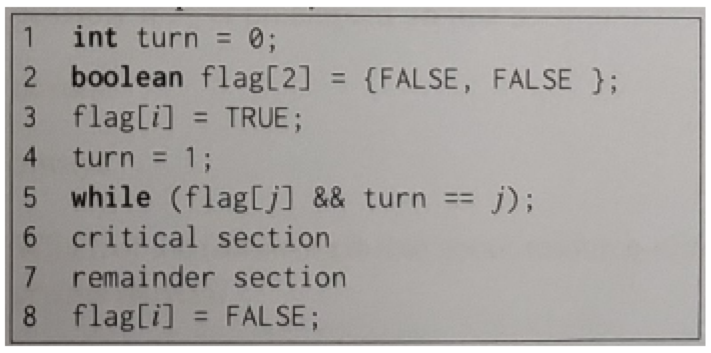
32. Which of the following correctly defines one of the scheduling criteria?
a) waiting time is the amount of time that a process spends in the wait queue.
b) throughput is the amount of time a process spends in the ready queue.
c) turnaround time is the amount of time from job sub- mission to completion.
d) CPU utilization is the amount of time a process spends in running state.
e) response time is the amount of time from job submis- sion to completion.

33. Which of the following is true about round-robin (RR) scheduling?
a) for a long time quantum, RR may suffer from starva- tion.
b) for a short time quantum, RR becomes similar to SJF.
c) for a short time quantum, RR becomes similar to FCFS. -> should be long time but not ans
d) for a short time quantum, RR may suffer from starva- tion.
e) for a long time quantum, RR becomes similar to SJF.

34. Continuing with the previous question, where can you put critical sections to eliminate the race con- dition?
a) around line 6 and another around line 12.
b) around line 3 and another around line 9.
e) around line 4 and another around line 10.
d) around line 5 and another around line 11.
e) around line 2 and another around line 10.

35. Which of the following is NOT a requirement of a critical section (CS)?
a) when no process is in its CS, a process wishing to enter its CS must not block indefinitely.
b) after a process has requested to enter its CS, other processes are allowed to enter their CS for a finite number of times.
c) there exists a limit on the number of times that other processes are allowed to enter their CS after a process has made its request to enter its CS.
d) a bound must exist on the number of times that other processes are allowed to enter their CS after the pro- cess has made a request to enter its CS.
e) at most one process is executing in its CS at any same time.

36. Which of the following is true about Peterson's solutions to the critical section problem? An excerpt is as follows for process P:



```
int turn = 0;
boolean flag[2] = {FALSE, FALSE };
flag[i] = TRUE;
turn = 1;
while (flag[j] && turn == j);
critical section
remainder section
flag[i] = FALSE;
```

a) progress is achieved because after the process R_{i} finishes its critical section, it always sets its own flag[i] = FALSE and turn is already == j.
b) Peterson's solution works on multicore architectures as long as you limit the number of processes to two.
c) mutual exclusion is achieved by the variable turn when flag[i]== flag[/] == TRUE
d) bounded wait is achieved because while loop's con- dition for i will always have an AND condition bro- ken by j if not already.
e) Peterson's solution assumes a non preemptive scheduler or else it will not work.

37. Which of the following is true about atomic TestAnd- Set, which works on a boolean variable named lock?
a) TestAndSet(lock) returns TRUE if the process suc- cessfully acquires the lock and sets lock to FALSE as a side effect.
b) to unlock, simply set lock = FALSE;
c) TestAndSet (lock) must evaluate to TRUE before the process can enter its critical section.
d) TestAndSet(lock) must evaluate to FALSE before the process can exit its critical section.
e) to lock, simply set lock = TRUE;

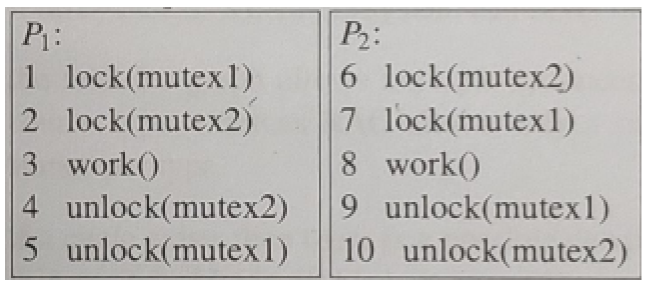
38. Which of the following is true about wait(S) and signal(S), where S is a semaphore?
a) wait (S) never blocks, but signal(S) may block.
b) both wait (S) and signal (S) may block.
c) wait(S) always blocks, but signal (S) never blocks.
d) wait (S) may block, but signal (5) never blocks.
e) wait(S) never blocks, but signal (S) always blocks.

39. What values should be on lines 4, 5, 7, 8?
a) empty, mutex, full, mutex
b) mutex, empty, full, mutex
c) empty, mutex, mutex, empty
d) full, mutex, mutex, empty
e) empty, mutex, mutex, full

40. In the Readers-Writers classical synchronization problem, two semaphores named rw_mutex and mutex are declared. What are their purposes?
a) rw_mutex allows either all writers or all readers to access the shared buffer.
b) rw_mutex allows at most one writer or unlimited number of readers to be accessing the shared buffer.
c) rw_mutex allows at most one writer to block out all other processes or the first reader to block all writers.
d) mutex allows at most one reader to be reading the shared buffer.
e) mutex allows at most one writer or one reader to be accessing the shared buffer.

41. Which of the following is not one of the necessary conditions of a deadlock?
a) circular wait for a resource held by another process.
b) each process is allowed to hold a resource while waiting to acquire another resource.
c) non preemptive process scheduling.
d) each process gets to hold a resource until voluntary release.
e) each resource is held by at most one process at a time.

42. Given two processes



P_1:	P_2:
1 lock(mutex1)	6 lock(mutex2)
2 lock(mutex2)	7 lock(mutex1)
3 work()	8 work()
4 unlock(mutex2)	9 unlock(mutex1)
5 unlock(mutex1)	10 unlock(mutex2)

When do they deadlock?
a) never if work() on line 3 and work() on line 8 are each wrapped in a critical section.
b) possibly if P is preempted between lines 1 and 2.
c) possibly if P_2 is preempted on line 8.
d) never.
e) always.

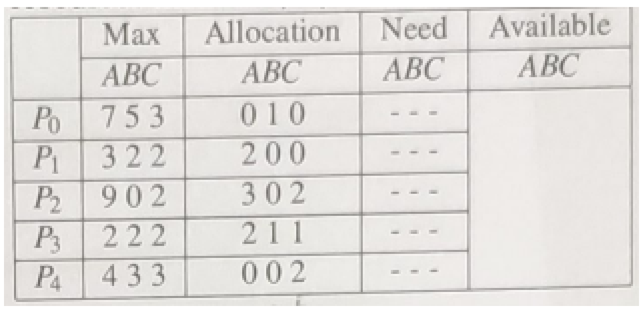
43. Which of the following is true about resource-allocation graphs (RAG)?
a) an edge from a process vertex to a resource vertex represents an assignment relationship. .
b) if a RAG contains a cycle and there are multiple instances per resource type, then the system may have a deadlock.
c) if a RAG contains a cycle, then the system has a deadlock.
d) if a RAG contains a cycle and there is only one in- stance per resource type, then the system may but does not always have a deadlock.
e) an edge from a resource vertex to a process vertex represents a request relationship. P

44. Which of the following is correct about deadlock avoidance?
a) it ensures that the system can break out of deadlock if it ever gets into one.
b) it ensures that the system never enters a deadlock. even if it enters an unsafe state..
c) it ensures that the processes never request resources in an unsafe order.
d) it ensures that the system never enters an unsafe state.
e) it ensures that no circular dependency exists in the resource requests among processes.

45. When using resource-allocation graph (RAG) for deadlock avoidance, request can be granted only if which of the following is true?
a) the RAG does not contain a cycle consisting of request edges and claim edges.
b) converting a request edge to assignment edge does not result in a cycle in the RAG.
c) converting a claim edge to an assignment edge does not result in a cycle in the RAG.
d) converting an assignment edge back to a claim edge does not result in a cycle in the RAG.
e) converting a request edge to a claim edge does not result in a cycle in the RAG.

46. Banker's algorithm uses the Safety Algorithm to find a safety sequence. If such a sequence is found, what does it mean?
a) it is a sufficient condition that the system is in safe state.
b) resources must be granted according to the safety sequence or else the system may result in deadlock.
c) it is a necessary but not sufficient condition that the system is in safe state.
d) resources must be granted according to the safety sequence or else the system enters an unsafe state.
e) it is neither a necessary nor a sufficient condition that the system is in a safe state.

47. In the Banker's algorithm, suppose you have three types of resources named A, B, and C,



	Max	Allocation	Need	Available
	ABC	*ABC*	*ABC*	*ABC*
P_0	7 5 3	0 1 0	- - -	
P_1	3 2 2	2 0 0	- - -	
P_2	9 0 2	3 0 2	- - -	
P_3	2 2 2	2 1 1	- - -	
P_4	4 3 3	0 0 2	- - -	

What are the Need values for process P_1?
a) 600
b) 011
c) 431
d) 743
e) 122

48. What is the available value, assuming there are 10 instances of A, 5 instances of B, and 7 instances of C?
a) 1057
b) 902
c) 433
d) 725
e) 332

49. Which of the following is true about the wait-for graph used for deadlock detection?
a) if no cycle exists then the system can never deadlock.
b) the wait-for graph allows multiple instances per re- source type, whereas RAG allows only a single in- stance per type.
c) if a cycle exists then there is a possible deadlock but it is an undecidable problem in general.
d) if no cycle exists then the system is in a safe state.
e) if a cycle exists then there is definitely a deadlock.

50. Which of the following is true about the Resource- Request algorithm when process P makes a request?
a) if Need,> Request; > Available, raise an exception because the process exceeds its maximum request.
b) when pretending to grant requested resources to P Allocation, gets incremented by Need.
c) if Request, > Available, then P, must wait.
d) if Request> Need, then P; must wait.
e) when pretending to grant requested resources to P Available, gets incremented by Request,

Quiz 5

1. What kind of binding is the logical address the same as the physical address?

a) Any binding using base-and-limit registers
b) Execution-time binding
c) Compile-time binding
d) Load-time binding
e) Any binding using translation lookaside buffer

2. Which of the following is false about load-time binding schemes?

a) The instructions issue the physical address instead of the logical address
b) The compiler must generate relocatable code
c) The base address register's value is given at load time
d) The loader substitutes the relocatable address with the base address plus the offset
e) The physical address is calculate by adding the base address to the logical address on every memory access

3. What is a problem with the combination of dynamic loading and static linking

a) It incurs high overhead because each memory access needs to check if the content is memory resident
b) It loads all code into memory even if it is never executed by the process
c) It wastes disk space because each executable needs a copy of the library in its disk image
d) Memory cannot be protected by dynamic loading
e) It could load multiple copies of the same code into memory

4. Which of the following is true about fragmentation in variable or fixed-sized contiguous memory allocation?

a) In variable-sized allocation, all fragmentations are internal
b) Both variable-sized and fixed size allocation can have both internal and external fragmentation
c) In fixed-size allocation, all fragmentations are internal
d) In fixed-sized allocation, all fragmentations are external
e) In variable-sized allocation, all fragmentations are external

5. Which of the following is true about compaction?

a) It requires compile-time address binding
b) It requires load-time address binding
c) Compaction may result in a more fragmented memory that it started with
d) Compaction may lose data if a lossy compression algorithm is used
e) It requires execution-time address binding

6. Which of the following is true about paging?

a) Paging eliminates internal fragmentation but still can have external fragmentation
b) Paging has neither internal nor external fragmentation
c) Paging is done by adding the relocation register to the logical address, and ensuring it is less than the page-table length register (PTLR)
d) Paging avoids external fragmentation but still can have internal fragmentation
e) A frame is a logical memory block and a page is a physical memory block

7. Let T denote the page table, and we denote the logical address and its corresponding physical address respectively as the concatenation of the page or frame number with the offset, $p_{logical} \cdot d_{logical}$ and $p_{physical} \cdot d_{physical}$. Which of the following is true?

a) $p_{logical} = T[p_{physical}]$
b) $p_{logical} \cdot d_{logical} = p_{physical} \cdot d_{physical}$
c) $d_{logical} = d_{physical}$
d) $p_{logical} \cdot d_{logical} = T[p_{physical} \cdot d_{physical}]$
e) $p_{logical} = p_{physical}$

8. What is NOT a benefit of demand paging compared to non-demand paging?

a) Provides a larger logical space than physically available
b) Does not load code into memory if it is never executed
c) Allows for more efficient process creation
d) Allows more programs to run concurrently
e) Provides better protection from other users

9. In a demand paging system, which one is a correct interpretation about the valid/invalid bit?

a) 'i' could mean either the page is within the process's address but nonresident or an invalid reference
b) 'v' means the page is in memory, 'i' means the page is outside the process's address space, and 'f' means the page is on disk (page fault)
c) 'v' could mean the page is memory-resident or it is non-resident but in the process's address space
d) 'v' means the page table entry contains a virtual address that needs to be translated
e) 'i' means the page is in memory

10. Is there a difference in how a page fault is handled compared to a regular interrupt?

a) The instruction that caused the page fault must be restarted after fault handling
b) No difference - a page fault is handled just like any other software interrupt
c) While handling a page fault all other interrupts must still be allowed due to the long time for disk access
d) While handling a page fault, no other interrupt is allowed or else page fault handling could be reentrant
e) A page fault is handled as a system call rather than as an interrupt

11. Which of the following is a correct definition?

a) A frame allocation algorithm decides which frame to allocate to which process
b) A frame allocation algorithm decides page to allocate to which frame
c) A page replacement algorithm decides which frame to replace
d) A page replacement algorithm decides how many pages to replace
e) A frame allocation algorithm decides how many pages to allocate to which frames

12. Which of the following is true about Belady's anomaly?

a) Demand paging can actually incur more page faults than non-demand paging
b) Second chance algorithm can actually incur more page faults than OPT
c) LRU algorithm could actually have more page faults than FIFO based
d) Adding more frames could actually cause more page faults
e) Stack algorithms can actually incur more page faults than FIFO based algorithms

13. Which of the following is true about Enhanced Second-Chance page replacement algorithm, assuming the tuple consists of (reference, modify) bits?

a) The tuple must be weighted by the access count in order to determine the priority for page replacement
b) (1, 0) is the best page to replace because it is recently used and clean, and the replacement cost is the lowest
c) (1, 1) is the best page to replace because it is recently used and modified, and replacing it will save paging cost
d) (0, 1) is the best page to replace because it is not recently used, and it is always good to write a copy to dist
e) A page with (0, 0) is the best page to replace because it is neither recently used nor modified

14. Which of the following is true about global vs local frame allocation upon page fault?

a) Global allocation tends to result in more thrashing
b) Global allocation has more consistent per-process performance
c) Local allocation has less predictable execution time due to possibly underutilized memory
d) global allocation is more common due to greater throughput and better utilization
e) Local allocation is more command due to greater throughput and better utilization

15. Which of the following is true about SLAB allocation?

a) Each cache has no external fragmentation but can still have internal fragmentation for the same reasons as paging
b) There is no fragmentation in each cache because it contains exactly the size of object needed
c) Each cache has no internal fragmentation but can still have external fragmentation when all holes are smaller than the requested object size
d) There is no fragmentation in each cache because all objects are powers of 2 in size (bytes)
e) A cylinder is the set of tracks on all platters at a given radius

16. Which of the following is true about the HDD mechanism?

a) Rotational latency is the time for platter to make one 360 degree rotation
b) The arm assembly moves the read/write heads on each platter individually
c) A sector is another word for a track
d) Seek time is the time for the sector to rotate under the read/write head
e) A cylinder is the set of tracks on all platters at a given radius

17. Which of the following disk scheduling algorithms may have starvation?

a) SCAN
b) SSTF
c) FCFS
d) C-SCAN
e) C-LOOK

18. Which of the following is true about NAND flash based storage when modifying one byte?

a) It needs to read out the entire block of pages to memory, modify the byte, erase the entire block, and write back the entire block
b) It is similar to RAM but with an extra step of erasing the word first before writing the modified value back
c) It is the same as RAM: load the word at the address to a register, modify, and store the word back to NVM
d) It needs to read out the page containing the byte, modify the byte in memory, erase the page, and write the page back
e) It needs to read out the page containing the byte, modify the byte in memory, and write the page back

19. What is the mean time to repair?

a) The time from when disk failure occurs to the time the replacement disk is working
b) The time from discovering failure to repair and restore the failed disk
c) The time from removing the failed disk and putting in the replacement disk
d) The time from when a sector first becomes defective to the time data is restored
e) The time from when a sector first becomes defective to the time a spare sector replaces the defective one

20. Which of the following is true about RAID?

a) It always stores the data on one disk and uses other disks to take snapshots to avoid disks failing at the same time
b) It improves reliability through redundancy and/or improves performance by stripping
c) It always mirrors disks but asynchronously to reduce wear and tear on drives
d) It always stores data one disk and keeps compressed images on other disks to avoid taking up too much space
e) It always mirrors disks synchronously to ensure disks do not fail at the same time

21. Using a processor cache as an analogy, what does each TLB entry contain in the cache tag and data?

a) segment number = cache tag, offset = data
b) page number = cache tag, frame number = data
c) physical address = cache tag, virtual address = data
d) frame number = cache tag, page number = data
e) virtual address = cache tag, physical address = data

22. What happens on a TLB miss?

a) it causes a page fault and the OS must swap in the page from disk
b) the OS flushes the entire TLB and forces the page table to be reloaded
c) the OS loads the page-table entry from memory in- stead of updating the TLB
d) the OS loads the page-table entry from memory into the TLB
e) the OS updates the page-table entry by setting the valid/invalid bit for corresponding TLB entry to 'i'

23. In a two-level hierarchical page table, the size of an inner page table is

a) depends on the size of the TLB
b) depends on only the size of the address space
c) the same as the size of a page
d) the same as the size of the outer page table
e) proportional to the size of the address space and in- versely proportional to the size of the outer page table

24. What is true about the way clustered page tables work?

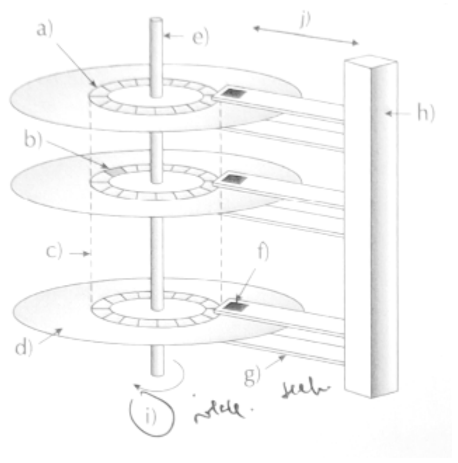
a) consecutive entries in the page table, called a cluster. are used instead of a linked list to record have hash collision pages that
b) multiple inner page tables are allocated on consecutive pages for better locality in hierarchical page tables
c) the most frequently accessed pages are grouped, or clustered, to consecutive page table entries to increase spatial locality of the page table access
d) each page table entry covers multiple pages rather than just one
e) multiple inner page tables share the same outer page table entry for smaller outer page table size in hierarchical page tables

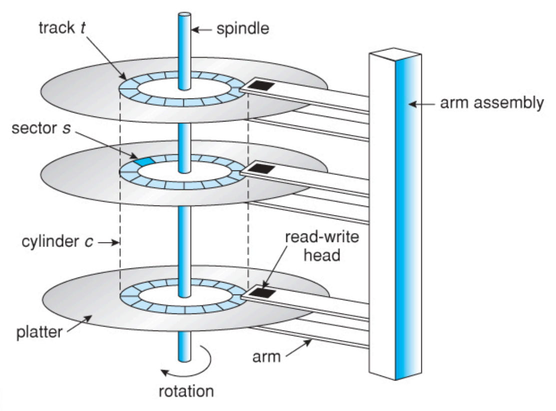
25. What is the purpose of the PTLR?

a) to save memory on the size of the page table when most entries are unused
b) to control the number of TLB entries to allocate to a given process
c) to prevent process from accessing uninitialized memory within a page
d) to control the size of the outer page table in a hierarchical page table
e) to prevent processes from accessing another process's page table

Quiz 6

Refer to the following figure for a magnetic disk:





1. What is c) called?
    a. Sector
    b. Platter
    [✓ c. Cylinder]
    d. Ring
    e. Track

2. What is the motion of j)?
    a. Peek
    b. Poke
    c. Ration
    [✓ d. Seek]
    e. read/write

3. What is h) called?
    a. Cylinder
    b. Spindle
    c. read/write head
    d. Disk arm
    [✓ e. Arm assembly]

4. What is the positioning time?
    a. The time of vertical motion of h) unit for the head be at the target height
    b. The time of j) motion for the head to be at the target radius
    c. The time of i) motion, and vertical motion of h) unit for the head to be on the target position
    [✓ d. The time of both i) motion and j) motion for the head to be on the target position]
    e. The time of i) motion for the head to be at the target angle

5. What is an issue with hard disk drives that is not an issue with NAND flash storage?
    a. Block erase
    b. Erase before write
    c. Minimum transfer unit of a page or sector
    [✓ d. Seek time and rotational latency]
    e. Wear-leveling

6. Which of the following disk scheduling algorithms is better for SSD than for HDD?
    a. LOOK
    b. STSF
    c. C SCAN
    [✓ d. FIFO]
    e. SCAN

7. Which of the following disk scheduling algorithms may have starvation?
    a. C SCAN
    b. FIFO
    [✓ c. STSF]
    d. LOOK
    e. SCAN

8. What is the difference between SCAN and LOOK disk scheduling algorithms?
    [✓ a. SCAN moves to both extremes of the disk cylinders, while LOOK moves only to the highest or lowest cylinders that are actually requested]
    b. LOOK has starvation but SCAN does not
    c. When SCAN reaches one extreme, it moves to the other extreme next
    d. When LOOK reaches one extreme, it moves to the other extreme next
    e. LOOK moves to both extremes of the disk cylinders, while SCAN moves only to the highest or lowest cylinders that are actually requested

9. Which RAID organization improves the mean time to data loss over a single hard drive?
    [✓ a. Mirroring as in RAID-1]
    b. RAID can only improve performance but not mean time to data loss
    c. Nonredundant stripping as in RAID-0
    d. Page-level stripping over multiple RAID disks
    e. Bit-level striping over multiple RAID disks

10. Assume you have consecutive sectors abedes where s is a spare and d is defective, what is the sequence of sectors after sector, after repair by sector slipping? Use u for marking a sector as unusable.
    a. Abccde
    b. Abcued
    c. Abeucd
    [✓ d. Abcude]
    e. Dabucd

11. In a personal computer, which of the following is attached directly to the PCIe bus directly?
    a. L2 and L3 cache of the CPU
    [✓ b. Graphics controller]
    c. Hard disk drive with SATA interface
    d. USB keyboard
    e. Monitor with HDMI interface

12. How would a SCSI device connect to PCIe?
    a. Via SCSI to USB converter
    [✓ b. Via host-bus adapter]
    c. Via UART
    d. Via ethernet adapter
    e. Connect directly since PCIE is backwards compatible with SCSI

13. How can the processor know if a device is busy?
    a. Write the device's control register
    b. Poll the device's IRO line
    c. Write the device's status register
    [✓ d. Read the device's status register]
    e. Read the device's control register

14. To control SATA hard drives, the processor would access the device registers that reside where?
    a. On the DMA controller connected to the PCIe bus
    b. On the processor in the I/O address space
    c. On the processor in the GPIO (0H-2FH) IRAM space
    d. On the processor in the special function register (SFR) space
    [✓ e. On the hard drive's controller]

15. What is interrupt chaining?
    a. Instead of nested interrupts, an ISR upon completion jumps to the next ISR instead of return and interrupt to save overhead.
    b. The processor offloads ISR operation to the device by sending it a chain of device-specific commands to the device.
    c. Instead of invoking one ISR multiple times, it jumps directly to the beginning again to reduce overhead.
    d. Instead of one long ISR, it is one short ISR for only the essential operations so it returns as quickly as possible and the rest of the code is scheduled as user tasks.
    e. Multiple devices share an IRQ line, and the ISR queries each device.

16. Which of the following is not a correct explanation of DMA?
    a. After setup, the device driver tells the DMA controller to start DMA transfer.
    b. The DMA controller must steal cycles from the processor when it is not accessing memory, but this may cause the processor to wait for memory longer.
    c. Upon completion, the DMA controller interrupts the processor.
    d. The DMA controller must transfer not only to memory but also update the processor's cache or else the processor will access the wrong data.
    e. To set up, the device driver tells the device controller to transfer data to memory at the starting address and the number of bytes.

17. Which of the following is a correct characterization of Unix I/O calls?
    a. seek() is synchronous
    b. put() is nonblocking
    c. get() is asynchronous
    d. select() is asynchronous
    e. read() is synchronous

18. What is vectored I/O?
    a. execute multiple ISRs to service multiple interrupts instead of taking them as separate interrupts.
    b. have one system call perform multiple I/O operations to save overhead.
    c. set up the interrupt vector to handle I/O operations instead of internal events such as exceptions.
    d. set up a DMA controller to execute multiple buffer addresses instead of taking multiple interrupts on completion of each buffer.
    e. divide a long ISR into a vector ("array") of code segments to avoid spending a long time inside an ISR.

19. Where is the file name of a file stored?
    a. in the inode of the file,
    b. in the hidden part of the file at the beginning of the file.
    c. in the file control block (FCB) of the file.
    d. in the directory structure on disk.
    e. in the hidden part of the file after the last data byte.

20. Which of the following Unix I/O calls is used to issue device-specific commands?
    a. write()
    b. signal()
    c. fcntl()
    d. ioctl()
    e. select()

21. (Preview Question) Which of the following structures is only in-memory but not on-disk?
    a. directory structure
    b. file control block
    c. volume control block
    d. boot control block
    e. system-wide open file table