Computer Science 284

Introduction to Operating Systems

Q: From the point of view of the calling thread, what actions occur from the time that a condition_wait() is called until the calling thread is allowed to resume execution?

A: 1) The calling thread is put to sleep;
   2) The mutex is unlocked;
   3) The calling thread is awakened by signal;
   4) The mutex is locked;
   5) The calling thread is in the mode "ready".
   6) Returns to the calling thread to resume execution

Q: What is the "lost wake-up" with respect to condition signals?

A: Condition signal is sent, but at that moment no thread is listening. Thus the signal gets lost. Therefore the thread waiting for this condition signal will be always waiting.

Q: What is a likely result of a "lost wake-up" having occurred?

A: Endless waiting.

Q: List a sequence of events that might result in a lost wake-up' if it were not programmed correctly. Number the events 1..N and be explicit as to where/when the signal 'gets lost'.

A: 1) Thread 1 tests condition;
   2) Thread 1 looses its time slice;
   3) Thread 2 sends a signal;
   4) Thread 2 looses its time slice;
   5) Thread 1 gets time slice and waits for the signal

Q: What does it mean that "condition signals do not pend"?

A: If no one is listening, the signal gets lost.

Q: A kernel level context switch may or may not be required to switch between user level threads (within the same process). What determines whether the kernel is involved or not when switching between threads?

A: Case 1: Several threads share one LWP. The kernel is not involved.
   Case 2: Threads do not share LWP. The kernel is involved.

Q: What does it mean to say that a library or a module is MT-Safe?

A: A library/module protects its global and static data with locks and can provide a reasonable amount of concurrency.

Q: Under Solaris 2 running on a uniprocessor, threads waiting for a lock always sleep rather than spin. Why is this true (and reasonable)?

A: In a uniprocessor system, only one thread can run at a time. If a thread is spinning for a lock, it is doing nothing but wasting CPU cycles while waiting for the lock to become available. Instead, it is better that they sleep (in the blocked queue) while waiting so that others can use the CPU. When the lock becomes available, OS will wake them up.

Q: Why is it always important to execute a "cond_wait()" inside a

critical section which is entered through a 'mutex lock' variable?
A: Without a mutex around the "signaler" and "waiter", there is a possibility that the signal might get lost.

Q: In dining philosophers problem, why examining and acquiring a fork is done inside a critical section? Explain by giving an example what may go wrong if critical section is not used.
A: Each fork is shared by two neighboring philosophers. If it is not handled inside a critical section, both philosophers may think that they acquired the same fork and start eating using the same fork. This situation leads to an incorrect simulation.

Q: In dining philosophers problem, which senario may lead to a deadlock situation?
A: If the simulation code is written in such a way that each philosopher first acquires the right fork and then attempts to acquire the left fork, a deadlock may occur. Because it is possible that philosophers may enter a circular wait situation in which each one holds the fork on the right and tries to get the fork on the left (which will never happen).

Q: True or False?
In Producer/Consumer problem access to shared buffer must be done in a critical section, but access to ``in'' and ``out'' pointers doesn't need to be done inside a critical section.
A: False. "in" and "out" are shared variables. They need to be accessed inside a critical section.

Q: True or False?
Even if mutual exclusion is not enforced on a critical section, results of multiple execution is deterministic (i.e. it produces the same results each time it runs).
A: False

Q: True or False?
If mutual exclusion is not enforced in accessing a critical section, a deadlock is guaranteed to occur.
A: False

Q: True or False?
It is impossible to implement mutual exclusion without using atomic operations.
A: True.

Q: True or False?
An acceptable solution for implementing mutual exclusion is to let the users disable interrupts before entering a critical section and enable them after leaving the critical section.
A: False

Q: True or False?
One of the solutions proposed for handling the mutual exclusion problem relies on the knowledge of relative speeds of processes/processors.
A: False. No solution should rely on such assumptions

Q: True or False?
In Readers/Writers problem, it is possible to write code which is functionally correct but may lead to the starvation of writers. However, it is impossible to write the code such that readers may starve instead of writers.
A: False

Q: What is the difference between starvation and deadlock?
A:
STARVATION: A condition in which a process is indefinitely delayed

because other processes are always given preference.

DEADLOCK: An impasse that occurs when multiple processes are waiting
for the availability of a resource that will not become available
because it is being held by another process that is in a similar state.

Q: What are the necessary conditions for a deadlock to exist.

A: a) mutual exclusion;
b) hold and wait;
c) no preemption
d) circular wait.

Q: True or False?
If all processes request all of the resources (in hardware and
software) that they need at once, then deadlock can never happen.
A: True

Q: What is a critical section i.e. what makes a section 'critical'?

A: In an asynchronous procedure of a computer program, a part that can
not be executed simultaneously with an associated critical section of
another asynchronous procedure.
In general, a code segment in which shared variables, shared file
descriptors (shared resources) are accessed is considered a critical
section.

Q: What is the difference between mutex_lock and mutex_trylock?

A: mutex_lock is a blocking call. If we try to lock a mutex that is already
locked by some other thread, mutex_lock blocks until the mutex is unlocked.
But mutex_trylock is a nonblocking function that returns if the mutex is
already locked.

Q: Under what conditions must a sema_t be initialized?

A: A semaphore is a non-negative integer count and is generally used to
coordinate access to resources. If threads are to be synchronized using
semaphores, semaphores must be initialized before use. The initial
semaphore count is set to the number of resources.

Q: What are the 3 possible dispositions that a process may specify with
respect to an interrupt?

A: 1) ignore signal;
2) run the default signal handler provided by the OS;
3) catch the signal and run the user's signal handler.

Q: True or False?
Different threads within a process may specify different
dispositions for a specific interrupt
A: True.

Q: Which of the following would not necessarily cause a process to be
interrupted?
(a) Division by zero
(b) reference outside user's memory space
(c) page fault
(d) accessing cache memory
(e) end of time slice
(f) none of the above

A: d

Q: An interrupt will pend unless a process specifies_____.
A: SIG_UNBLOCK.

Q: An interrupt will get serviced immediately if a process
specifies_____
A: SIG_UNBLOCK.

Q: Using signal(...), show the program parts necessary to cause a process to print a message the 1st time that the keyboard operator presses ^C (and further ^Cs would exit the program). For the same program, show/describe what has to be changed to get the message N times and then exit on the (N+1)th time? (assume that N is hard coded).

A: See the homework.

Q: What is a sigset_t? How is a variable of this type set/modified? In what ways can a variable of this type be used after it has been assigned an appropriate value(s)?

A: sigset_t specifies a signal set. sigemptyset(), sigfillset(), sigaddset(), and sigdelset() are used to set or modify a sigset_t variable. sigprocmask() or thr_sigsetmask() changes or examines the caller's signal mask by using sigset_t variables.

Q: What functions are used from within a process to send an interrupt to a process?

A: sigsend()

Q: When reading from a pipe, what causes the kernel to send EOF to the reader?
A: When the buffer is empty and the write end of the pipe has been closed.

Q: True or False?
We can use one pipe to implement two way communication between processes.
A: False. Need two pipes to do that. Pipes are unidirectional

Q: Why can parent/child processes communicate via unnamed pipes but 2 unrelated processes can not?

A: Child process is an exact copy of the calling process (parent process). The child process copies file descriptor with parent process and inherits almost all attributes from the parent process. The two unrelated process don't know the attributes of each other.

Q: If your program contains a race condition, what does that mean?

A: Race conditions occur when shared data is accessed by different processes without explicit synchronization. Programs containing races can behave in ways unexpected by their programmers.

Q: How can the code segment "if(i < y) cout << foobar(i,y);" be made to behave as if it were atomic? Answer the question by recoding the segment.

A: mutex_t m;

```
   mutex_lock( &m );
   if( i < y )
      cout << foobar( i, y );
   mutex_unlock( &m );
```

Q: What are the three popular strategies for allocating free memory
blocks to processes in dynamic memory partitioning? Explain briefly
how each strategy works.
A: First-fit: chooses the first free block in the list that is large
enough for the request.
Best-fit: chooses the free block that is closest in size to the
request.
Next-fit: chooses the first free block that is large enough for
the request and comes after the ``Last Allocated Block''
in the list.
Q: Give an example drawing of a partially allocated memory and
appropriate pointers in which each strategy ends up allocating
a different free memory block to satisfy a memory request for 16 MB.
Indicate clearly which strategy allocates which memory block.
A: Refer to the example in Chapter 7 slides.

Q: What interrupt is created when a desired frame is not currently resident
in RAM?

A: Page fault trap.

Q: How does the hardware 'know' that a desired frame is not currently
resident in RAM?

A: Valid bit.

Q: What precisely does it mean if the 'dirty bit' is set for a frame?
A: The frame has been modified.

Q: What is 'good' vs. 'bad' program locality?
A: 'Good' locality means that the process executes in clustered pages.
'Bad' locality means that the process executes in scattered pages.

Q: Explain when/how internal fragmentation may occur.
A: Paging system uses fixed-sized pages. Thus internal fragmentation may
occur.

Q: Explain when/how external fragmentation may occur.
A: Segmentation system breaks up the memory space into variable-sized pieces.
Thus, external fragmentation may occur.

Q: You will be given a 'claim matrix', an 'allocation matrix' and a
'resource vector' for a set of processes. You will then be given a
resource request from one of the processes. Should the request be
granted? and if yes, give a <safe sequence>.

Claim Matrix

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

Allocation Matrix

	R1	R2	R3
P1	1	0	0
P2	5	1	1
P3	2	1	1
P4	0	0	2

Resource Vector

R1	R2	R3

9 3 6

Process P(i) requests N units of resource R(j). Is this a 'safe' request?
Give a <safe sequence>.

A:

Need Matrix

	R1	R2	R3
P1	2	2	2
P2	1	0	2
P3	1	0	3
P4	4	2	0

Yes, <P2, P1, P3, P4>

Q: Describe the differences between a global allocation scheme and the working-set model for allocation.

A: Global replacement allows a process to select a replacement frame from the set of all frames, even if that frame is currently allocated to some other process; one process can take a frame from another.

The working set model assumes that processes execute in localities. The working set is the set of pages in the current locality. Accordingly, each process should be allocated enough frames for its current working set.

Q: Comparing global allocation vs. working set allocation, which would be more adversely affected by a program with 'bad' locality? and WHY would that be true?

A: Working set allocation would be more adversely affected by a program with 'bad' locality. This is because the program with 'bad' locality has poorly defined working sets.

Q: What is the "largest" program that could execute on a machine with a 24-bit virtual address?
A: 2^24 byte

Q: The address contained in a page table entry <PTE> is (physical | logical).
A: physical

Q: The address contained in a TLB entry <PTE> is (physical | logical).
A: physical

Q: List at least 3 flags that are contained in a PTE.
A: Valid bit, Reference bit, Dirty bit.

Q: Define hit-ratio in a memory management context.
A: The percentage of times that a page number is found in the associative registers is called the hit ratio.

Q: True or False?
If a virtual page number X generates a miss in the TLB (Translation Lookaside Buffer), then the translation for X is guaranteed to be found in the Page Table Entry.
A: False. Not necessarily. It can be in disk

Q: True or False?
It is possible that page tables are stored in virtual (secondary) memory.
A: True. when multi-level paging schemes are used

Q: True or False?
In a virtual memory system with paging, page size must be large enough to offset the high cost of page faults.

A: True

Q: True or False?
   The Least Recently Used (LRU) page replacement strategy is based on
   the principle of spatial locality (locality in space) as opposed to
   temporal locality (locality in time).
A: False. LRU strategy is based on the temporal locality.

Q: True or False?
   Consider {\bf clock policy} for page replacement, a newly arrived
   page (i.e. just swapped in) will not get replaced before the clock
   pointer makes two full rotations in the circular buffer of
   candidate page frames.
A: True. use=1 at arrival, use=0 after the first rotation of the clock
   pointer. if use=0 remains true after the second rotation, it may
   get replaced; otherwise, it stays on.

Q: True or False?
   In a virtual memory system with paging, you can run a program whose
   size is larger than the size of Main Memory.
A: True.

Q: True or False?
   A unix socket is used for communication between processes running
   on the same machine. On the other hand, an internet socket can not
   be used for communication between processes running on the same machine.
A: False.Internet Socket can be used for processes running on the same
   machine as well as different machines

Q: True or False?
   The function call listen(soc,N), allows N clients to wait to be connected.
A: True

Q: True or False?
   If clients are connected to a server through "connect()" and
   "accept()" calls and the server calls "listen(soc,2)" before
   "accept()", then at most 2 clients can get connected to the server
   at anytime.
A: False. listen() determines the size of the wait queue before the
   connections take place, not the max. number of clients that can get
   connected.

Q: True or False?
   connect() returns a file descriptor associated with the new connection.
A: FALSE

Q: True or False?
   In RAID (Redundant Array of Independent Disks) Level 1, every disk
   in the array has a mirror disk that contains the same data.
A: True.

Q: Which of the following strategies is not used in a Disk Scheduling
   Algorithm?
  (a) First in first out (FIFO)
  (b) Last in first out (LIFO)
  (c) Shortest service time first (SSTF)
  (d) Longest service time first (LSTF)
  (e) Back and forth over disk (SCAN)

A: d.

Q: Which one of the following is not among the set of events that may
   take place between the time a page fault occurs and the time the
   faulting process resumes execution?
   (a) OS blocks the process and puts it into a wait queue.
   (b) One of the processes in the ready queue is selected to run.
   (c) A DMA is initiated to load the page from disk into main memory
   (d) A page replacement strategy is used to find a page frame to

```
       load the new page
   (e) Page table is updated to reflect the change.
   (f) none of the above

A: f

Q: Which one of the following is not among the set of events that may
   take place between the time a page fault occurs and the time the
   faulting process resumes execution?
   (a) OS blocks the process and puts it into a wait queue.
   (b) One of the processes in the ready queue is selected to run.
   (c) A DMA is initiated to load the page from disk into main memory
   (d) The last page that the faulting process was executing is
       replaced with the newly loaded page.
   (e) Page table is updated to reflect the change.
   (f) none of the above

A: d


Q: True or False?
   While DMA (Direct Memory Access) is taking place, processor is free
   to do other things. The processor is only involved at the beginning
   and end of the DMA transfer.
A: True

Q: True or False?
   DMA uses "cycle stealing" to transfer data on the system bus. Each
   time cycle stealing is used, CPU is interrupted.
A: False. The only interrupts that occur during DMA transfer is at the
   beginning and at the end of DMA (no interrupts in between).

Q: How does the kernel 'know' where on disk the desired information is for
a non-resident frame?

A: Swap map.


Q: Describe what demand paging means.
A: The technique of only loading virtual pages into memory as they are
accessed is known as demand paging. If the demand pages are not in memory,
a page fault trap happens, and the operation system swaps them in.

-> Must know Figures 8.3 - 8.8 (except 8.6) in detail

Q:  Explain what the following C calls do both when the call is
    successful and when it is unsuccessful.
    1.  socket( AF_INET, SOCK_STREAM, 0 )
    2.  bind(sd, (struct sockaddr*)&server_addr, sizeof(server_addr))
    3.  accept( sd, (struct sockaddr*)&client_addr, &client_len )
A:
   1. creates an internet socket and returns the socket descriptor.
      If the call fails, it returns -1.
   2. Binds the definition of a socket (socket descriptor) to a port number.
      If the call fails, it returns -1.
   3. Blocks execution until a client connection is received. When
      that happens, it returns a descriptor for the connection.
      If the call fails, it returns -1.

Q: Which one of the following is not among the 7-layers defined for
   ISO Open Systems Interconnect (OSI) model ?
   (a) Application
   (b) Routing
   (c) Transport
   (d) Data Link
   (e) Physical

A: b
```

==========
Please fill in the blanks for the following questions:

Q: If a desired page frame is not currently resident in RAM,
   _____ occurs.
A: A Page Fault

Q: Since paging system uses _____-sized pages,
   _____ fragmentation may occur.
A: fixed size; internal

Q: If a memory management system uses dynamic partitioning,
   _____ fragmentation may occur.
A: External

Q: _____ is a form of I/O in which a special module
   controls the exchange of data between main memory and an I/O
   device. During this I/O transfer, CPU is free to do other computation.
A: DMA (Direct Memory Access)

Q: _____ techniques guarantee that a deadlock will
   not occur at all. This is achieved by assuring that one of the
   necessary conditions for deadlock is not met.
A: Deadlock Prevention

Q: A portion of the operating system that includes the most heavily
   used portions of software and, generally, maintained permanently in
   the main memory is called _____.
A: Kernel

Q: The least recently used (LRU) page replacement strategy is
   based on the principle of _____ as opposed to
   _____.
A: temporal locality; spatial locality

Q: The top four levels in the memory hierarchy, starting with the
   fastest, are: _____; _____; _____;
   _____
A: Registers; cache memory; RAM; Disk.

Q: Swapping out a piece of a process (i.e. pages of a process) just
   before that piece is needed is called _____
A: Thrashing

==========

Q. Miscellaneous questions related to the VIDEO: TRIUMP OF THE NERDS

1. _____ and _____ are generally credited
   with the invention of C/Unix?
2. _____ and _____ started Microsoft in
   19_____.
3. What corp./laboratory may fairly take credit for inventions like the
   mouse, windows, pull-down menus etc.?
4. _____ and _____ co-founded Apple.
   _____ then started NeXT, and is now the CEO of Pixar?
5. What company purchased NeXT and their OS NExT Step?
6. What did Steve Jobs see while visiting PARC that inspired him to build
   a different kind of computer?

What did he see that he completely ignored?
What was the 1st computer that he built based on this inspiration (that flopped)?
What was the 2nd one that didn't flop?

7. What 'product' got Microsoft into the microcomputer software business?
8. What lucky event got Microsoft into the operating system market?

Gary Kildall didn't eagerly pursue IBM when they requested a new OS. His wife and attorney would not sign a nondisclosure agreement. Bill Gates of Microsoft saw this as an opportunity and jumped in.

9. MS/DOS was 90% derived from a predecessor product named _____
   which was written by _____
   and owned by _____.
   which in turn had been cloned from _____
   written by _____
10. What is a 'killer application'?
11. What was the killer app for the Apple II?
12. What was the killer app for the IBM PC?
13. What was the killer app for the Apple MacIntosh?
14. Why didn't IBM create their own OS for their 1st PC?
15. Who 'should have' sold IBM their operating system for the 1st IBM PC?

Gary Kildall of Digital Research

16. What was the one part of the 1st IBM PC that was proprietary (that Compaq had to later reverse engineer)?
17. Why did IBM decide to build the PC using 'open architecture'?

To save time, instead of building a computer from scratch, IBM initially decided to buy PC components off the shelf and assemble them -- in IBM terms, this was called an 'open architecture'. IBM made some changes to this initial decision.

What was the almost immediate result of IBM having made that decision?

IBM had to buy the OS and other software from other companies as well.

18. What was IBM's motivation for designing/building PS-2/OS-2?

IBM planned to steal the market from Gates with a brand new OS called OS/2.

19. What person _____ what company _____ built the 1st commercially available personal computer in 1975?

20. Gordon Moore is one of the _____ founders.
21. World's first personal computer, _____, was designed by _____ and was introduced in 19___
22. The first mass market PC company is _____.

==========
ADDITIONAL QUESTIONS
==========

Q: True or False?
Buddy Strategy always allocates memory in chunks of size power of two and uses a data structure based on a binary tree.
A: True.

Q: What does an Internet Protocal do?
A: 1. Provides a naming scheme which uses a uniform format for host addresses
2. Provides a delivery mechanism by defining a standard packet format.

Q: True or False?
Sockets are bidirectional communication ports in UNIX. Once a

socket is created, it can be bound to an internet port using socket() call.
A: False. First statement is true but the 2. statement is false. Sockets can be bound to an internet port using bind() call.

Q: True or False?
There is only one internet port in each networked host.
A: False. There are many internet ports in each host; some are reserved by the OS.

Q: True or False?
The UNIX call {\it listen(soc,N)} allows only N clients to be connected to a socket at any time.
A: False. N species the length of the wait queue for the clients who are waiting to be connected.

Q: True or False?
The two lowest layers in the 7-layer ISO Open Systems Interconnect (OSI) model are Physical and Data Link layers and their primary function is to implement the TCP/IP protocol.
A: False. First part of the statement is true but the second part is false, because Transport layer (which is the 4. layer from the bottom) implements TCP/IP.

Q: Fill in the blanks.
The two lowest layers in the 7-layer ISO Open Systems Interconnect (OSI) model are _____ and _____ layers and their primary function is to provide _____ and _____ .
A: Physical; Data Link; signaling technology; frame management.

Q: Fill in the blanks.
Two transport protocols, _____ and _____, are defined and handled at the Transport Layer.
A: Transmission Control Protocol (TCP); User Datagram Protocol (UDP)

Q: What are the possible goals that any scheduling policy might try to accomplish (list at least three)?
A: To improve:
- response time
- Turnaround time (TAT)
- Throughput
- Processor Efficieny

Q: True or False?
Long-Term scheduler controls the degree of multiprogramming
A: True

Q: True or False?
Among the three scheduling disciplines (long-term, medium-term, and short-term), long-term scheduler executes most frequently.
A: False. Short-term scheduler (dispatcher) executes most frequently.

Q: True or False?
Among the short-term scheduling policies, feedback policy penalizes jobs that have been running longer.
A: True