

Fall 2020 - COM SCI111-1 - REIHER

Started on Friday, 30 October 2020, 3:27 PM PDT**State** Finished**Completed on** Friday, 30 October 2020, 5:27 PM PDT**Time taken** 2 hours**Grade** 74.25 out of 99.00 (75%)**Question 1**

Correct

3.00 points out of 3.00

Which of the following are necessary parts of an Application Binary Interface?

- a. Information about operation semantics
- b. Stack frame structure
- c. Register conventions
- d. A complete list of possible error return codes
- e. Load module format



Your answer is correct.

The correct answers are:

Register conventions,

Stack frame structure,

Load module format

Question 2

Incorrect

0.00 points out of 3.00

Which of the following is an advantage of implicitly (vs. explicitly) determining which memory segment an address belongs to?

- a. It makes relocation easier
- b. It makes garbage collection easier
- c. It allows variable numbers of segments
- d. It supports sparse segments
- e. It allow more flexibility in the size of segments



Your answer is incorrect.

The correct answer is:

It allow more flexibility in the size of segments

Question 3

Correct

3.00 points out of 3.00

Match each type of process memory segment with a set of its characteristics that best describes it.

Stack segment	Readable, writeable, extendable based on process behavior	✓
Code segment	Fixed size, sharable, and executable	✓
Data segment	Readable, writeable, extendable by a system call	✓

Your answer is correct.

The correct answer is:

Stack segment → Readable, writeable, extendable based on process behavior,

Code segment → Fixed size, sharable, and executable,

Data segment → Readable, writeable, extendable by a system call

Correct

3.00 points out of 3.00

Which of the following are differences between the Linux/Unix `brk()` and `malloc()` calls?

- a. `malloc()` handles memory allocation, while `brk()` deals with process destruction
- b. `brk()` is a library function, while `malloc()` is a system call
- c. `brk()` is a system call, while `malloc()` is a library function ✓
- d. `malloc()` handles memory management, while `brk()` is used when a process wants to yield
- e. `brk()` increases the size of the heap, while `malloc()` doesn't ✓

Your answer is correct.

The correct answers are:

`brk()` is a system call, while `malloc()` is a library function,

`brk()` increases the size of the heap, while `malloc()` doesn't

Question 5

Correct

3.00 points out of 3.00

What is the main benefit of buddy allocation in memory management?

- a. It makes memory compaction easier
- b. It allows easy coalescing on memory free ✓
- c. It solves the problem of contiguous allocation
- d. It allows for easy relocation
- e. It avoids internal fragmentation

Your answer is correct.

The correct answer is:

It allows easy coalescing on memory free



Which of the following is an example of the desirable property of coherence when designing operating system modules?

- a. Maintaining a stable interface for process creation over the course of multiple OS upgrades
- b. Putting the code to manage a particular buffer pool in the same module as the code for general memory allocation
- c. Providing both a user mode and system mode stack for each process
- d. Putting the allocation and free routines for memory management in the same module
- e. Providing applications with a file system interface rather than a storage device interface



Your answer is correct.

The correct answer is:

Putting the allocation and free routines for memory management in the same module

Question 7

Correct

3.00 points out of 3.00

In the context of operating systems, what does the term "dispatching" mean?

- a. Terminating a process
- b. Moving all of a process' RAM contents temporarily onto disk
- c. Choosing which process to run next
- d. Preparing a CPU or CPU core to run a particular process
- e. Sending data from one process to another



Your answer is correct.

The correct answer is:

Preparing a CPU or CPU core to run a particular process



In typical use, what is the relationship between Unix/Linux fork and exec system calls?

- a. The parent calls fork and both the parent and child call exec
- b. The parent calls fork and the child calls exec
- c. The parent calls exec and the child calls fork.
- d. The parent and child both call fork and neither calls exec

✔ That's how this OS creates a new process running a different program.

Your answer is correct.

The correct answer is:

The parent calls fork and the child calls exec

Question 9

Incorrect

0.00 points out of 3.00

What is the purpose of partition registers?

- a. Partition registers are used to minimize internal fragmentation. ✘ These registers merely indicate the start and end of partitions. They have nothing to do with whether the partition is exactly the size requested, which is the question that determines if internal fragmentation occurs.
- b. Partition registers are used to determine if a particular main memory partition is free or allocated.
- c. Partition registers determine whether the address a process requested in in RAM or secondary storage.
- d. Partition registers determine whether a process has issued an address in the range allocated to it.
- e. Partition registers are used to support relocation of partitions

Your answer is incorrect.

The correct answer is:

Partition registers determine whether a process has issued an address in the range allocated to it.



Which of the following elements of process state would typically **not** be kept in a process descriptor?

- a. Process priority
- b. Number of CPU seconds that the process has run so far ✘
- c. Process ID number
- d. The process' code ✔ Kept in a memory segment owned or shared by the process.
- e. The process stack ✔ The stack is kept in a memory segment belonging to the process.

Your answer is partially correct.

You have selected too many options.

The correct answers are:

The process stack,

The process' code

Question 11

Incorrect

-0.75 points out of 3.00

Which of the following is an element of a running process, but not of a program on a persistent storage device?

- a. Static libraries
- b. A data area
- c. The code
- d. The stack
- e. The ISA used ✘ That's explicitly in the program, but only implicitly in the process

Your answer is incorrect.

The correct answer is:

The stack

Correct

3.00 points out of 3.00

For which of the following types of jobs will round robin scheduling be most likely to work well?

- a. Data mining on huge data sets
- b. Periodic compaction of data on a persistent storage device
- c. Hard real time jobs
- d. Word processing on a shared server machine
- e. Soft real time jobs

✓ Lots of interaction, little computation will work well with round robin.

Your answer is correct.

The correct answer is:

Word processing on a shared server machine

Question 13

Partially correct

2.25 points out of 3.00

A carelessly written program can corrupt which of the following things in memory by running its own instructions, without system calls?

- a. Its code segment
- b. Its shared libraries
- c. The contents of its registers
- d. The data segment of a forked child process
- e. Its own stack
- f. Its process descriptor
- g. Its own data segment

✓

✓

✗ Not writeable by the process

✓

Your answer is partially correct.

You have selected too many options.

The correct answers are:

Its own stack,

Its own data segment,

The contents of its registers



Which of the following types of memory segments are typically shared by multiple processes?

- a. Stack segments
- b. Shared libraries
- c. Segments containing process code
- d. The segment containing the operating system's code
- e. Dynamic libraries
- f. Static libraries



Your answer is partially correct.

You have correctly selected 2.

The correct answers are:

The segment containing the operating system's code,

Segments containing process code,

Shared libraries

Question 15

Correct

3.00 points out of 3.00

Which of the following statements is true of trap handlers in a typical operating system?

- a. An application can create its own trap handlers
- b. Trap handlers run in user mode
- c. They are specified at boot time
- d. Each process has its own copy of trap handlers
- e. They are a part of the OS kernel



Your answer is correct.

The correct answers are:

They are specified at boot time,

They are a part of the OS kernel



Which of the following is a major difference between a trap and an interrupt?

- a. Interrupts originate internally in the CPU, while traps originate outside the CPU
- b. Traps alter the program counter on the processor, while interrupts do not
- c. Traps always result in switching from a process running to OS code running, while interrupts never do
- d. Traps are essential for running preemptive schedulers, while interrupts are not
- e. Traps are issued by processes, while interrupts are issued by peripheral devices

✘ Other way around

Your answer is incorrect.

The correct answer is:

Traps are issued by processes, while interrupts are issued by peripheral devices

Question 17

Correct

3.00 points out of 3.00

Which of the following operations are done as part of linkage editing?

- a. The code segment is copied into a process' memory
- b. Edit relocation entries in the load module to set them to the correct addresses ✓
- c. All code and data elements of the program are placed in a single virtual address space ✓
- d. The OS determines if a copy of any shared library used is already in memory, loading it if not
- e. External references in the object code are located in specified libraries ✓

Your answer is correct.

The correct answers are:

External references in the object code are located in specified libraries,

All code and data elements of the program are placed in a single virtual address space,

Edit relocation entries in the load module to set them to the correct addresses



What are the advantages of a bit-map free-list.

Select one or more:

- a. it is relatively easy to find contiguous free space
- b. they can also be used to represent which blocks have been allocated to a file
- c. they eliminate the need for garbage collection
- d. the allocate and free operations are very efficient
- e. they are very small



Your answer is partially correct.

You have correctly selected 2.

The correct answers are: they are very small, it is relatively easy to find contiguous free space, the allocate and free operations are very efficient

Question 19

Correct

3.00 points out of 3.00

Which of the following are characteristics of multi-queue scheduling?

Select one or more:

- a. Service within each queue is Round-Robin ✓ All processes within a single queue are deemed to have comparable scheduling characteristics.
- b. A task that experiences many Time-Slice-End preemptions will be moved to a longer quantum queue ✓ The task obviously needs longer (and probably less frequent) time slices.
- c. Regular blocking for frequent I/O moves a process to a lower priority queue
- d. The goal is for every task in every queue to consume its entire time-slice
- e. All queues receive equal CPU shares
- f. Controlling queue transitions by parameters enables *Mechanism/Policy* separation ✓ Queue transitions based on behavior is a mechanism. The parameters (max blocks, max time-slice ends, max-waiting time, ...) embody the policies.

Your answer is correct.

The correct answers are: Service within each queue is Round-Robin, A task that experiences many Time-Slice-End preemptions will be moved to a longer quantum queue, Controlling queue transitions by parameters enables *Mechanism/Policy* separation

Correct

3.00 points out of 3.00

Which of the following helps combat external fragmentation?

Select one:

- a. Fine-grained segmentation
- b. Protection bits
- c. Coarse-grained segmentation
- d. Negative-growth support
- e. Memory compaction



Your answer is correct.

The correct answer is: Memory compaction

Question 21

Correct

3.00 points out of 3.00

Which of the following are common errors in code that uses dynamically allocated memory?

Select one or more:

- a. continuing to use allocated memory after it has been freed
- b. calling malloc with the wrong alignment for the current Instruction Set Architecture.
- c. using heap allocation where stack allocation would have sufficed
- d. Using signed comparisons on pointers.
- e. freeing the same memory multiple times
- f. dereferencing a pointer that has not yet been initialized
- g. failing to free allocated memory
- h. failure to cast malloc'd memory to the proper type



Your answer is correct.

The correct answers are: failing to free allocated memory, continuing to use allocated memory after it has been freed, freeing the same memory multiple times, dereferencing a pointer that has not yet been initialized



What is the convoy effect?

Select one:

- a. In a FIFO system, one long running job can seriously delay shorter jobs' completion ✓
- b. Using SJF scheduling to run a group of short jobs before running a long one
- c. Once a schedule of jobs has been set up, one can efficiently move from one to the next
- d. Gaining performance benefit by scheduling jobs that use the same memory segments on the same CPU of a multicore system
- e. High priority jobs keep interrupting the processing of a low priority job, increasing its turnaround time

Your answer is correct.

The correct answer is: In a FIFO system, one long running job can seriously delay shorter jobs' completion

Question 23

Partially correct

1.50 points out of 3.00

Which of the following memory management approaches is likely to suffer from external fragmentation?

- a. Fixed partition allocation
- b. Management of sectors on HDDs
- c. Dynamic paging
- d. Dynamic partition allocation ✓ There may be free space of useless sizes between dynamic partitions.
- e. Buddy allocation
- f. Slab allocation ✗ Slab allocation produces chunks of fixed size blocks on known utility, so no external fragmentation.

Your answer is partially correct.

You have correctly selected 1.

The correct answers are:

Dynamic partition allocation,

Buddy allocation



Process A performs an flock() on an open file descriptor. Process A then forks off processes B, C, and D, without releasing the lock or closing the file. Which of the following statements is true?

Select one:

- a. The lock will only be released when process A releases it.
- b. If any of A, B, C, or D release the lock, it is released for all of them. ✓
- c. Each process has its own lock, which it can release independently of the others.
- d. Only A (not B, C, or D) holds the lock.
- e. The lock will be released if any of the processes exit.

Your answer is correct.

The correct answer is: If any of A, B, C, or D release the lock, it is released for all of them.

Question 25

Correct

3.00 points out of 3.00

Which of the following will free memory held by a program in a Unix/Linux style system?

Select one or more:

- a. Explicitly making the proper library call frees automatic memory
- b. Exiting a subroutine frees automatic memory ✓
- c. Terminating a program frees automatic and heap memory ✓ Even static memory, as well
- d. Exiting a subroutine frees both automatic and heap memory
- e. Explicitly making the proper library call frees heap memory ✓
- f. Exiting a subroutine frees heap memory
- g. Removing the last reference to a piece of memory frees it

Your answer is correct.

The correct answers are: Exiting a subroutine frees automatic memory, Terminating a program frees automatic and heap memory, Explicitly making the proper library call frees heap memory



Which of the following kinds of memory management approaches will experience internal fragmentation?

- a. Slab allocators
- b. Fixed partition allocation ✓ The poster child for internal fragmentation.
- c. Dynamic partition allocation
- d. Sector-based HDD memory management ✓ Sectors are a fixed size, so internal fragmentation may occur
- e. Buddy allocation ✓ Allocates only in powers of 2, so generally internal fragmentation is possible.
- f. Dynamic paging

Your answer is partially correct.

You have correctly selected 3.

The correct answers are: Dynamic paging,
Fixed partition allocation,

Buddy allocation,

Sector-based HDD memory management



Which of the following are characteristics of preemptive scheduling?

Select one or more:

- a. improved response time
 - b. minimum time-slice length is determined by the timer granularity
 - c. improved turn-around time
 - d. incompatible with priority-based scheduling
 - e. commonly used in hard real-time systems
 - f. increased overhead
- ✓ shorter time-slices reduce the length of time a process will have to wait to run
- ✓ preemption adds additional context switches, which take time to perform and result in cache loss.

Your answer is partially correct.

You have correctly selected 2.

The correct answers are: improved response time, increased overhead, minimum time-slice length is determined by the timer granularity

Question **28**

Correct

3.00 points out of 3.00

What is the purpose of a `pthread_join()` call?

Select one:

- a. To query a thread's status
- b. To wait for a thread to exit
- c. To cause a thread to exit
- d. To combine the results of two threads of one process
- e. To create a new thread



Your answer is correct.

The correct answer is: To wait for a thread to exit



How is segmentation different from base/limit relocation?

Select one or more:

- a. segmentation allows an address space to be built up from non-contiguous pieces of physical memory ✓
- b. segmentation allows the creation of an address space with holes ✓
- c. segmentation requires more relocation registers
- d. the compiler/linkage editor does not need to know how many segments there are
- e. segment addresses are generated by adding a positive offset to the (base) lowest address in the segment ✗ Segments can grow upwards or downwards, so offsets may be positive or negative.
- f. segmentation allows read-only and read/write access to be provided to different parts of a single segment ✗ all data in a single segment is treated the same way

Your answer is partially correct.

You have selected too many options.

The correct answers are: segmentation allows the creation of an address space with holes, segmentation allows an address space to be built up from non-contiguous pieces of physical memory, segmentation requires more relocation registers

Question 30

Correct

3.00 points out of 3.00

What is the best description of *starvation* in the context of operating systems?

Select one:

- a. A problem solvable by priority scheduling
- b. A potential problem with all non-preemptive scheduling algorithms
- c. potentially unbounded waiting times ✓
- d. A potential problem with all preemptive scheduling algorithms
- e. Inability to handle a resource request because of resource fragmentation

Your answer is correct.

The correct answer is: potentially unbounded waiting times



Which of these is the best definition of thrashing?

Select one:

- a. A situation where the memory demands of the running processes exceed the amount of physical memory available ✓
- b. A situation where one or more processes are halted and totally swapped to disk to improve overall paging performance
- c. Behavior of a paging algorithm that scans the full set of pages without finding one suitable for replacement
- d. Behavior of a paging algorithm under a particular workload that always results in replacing a page that will be referenced in the near future
- e. A situation where the memory manager chooses to kill one memory intensive process to improve overall paging performance

Your answer is correct.

The correct answer is: A situation where the memory demands of the running processes exceed the amount of physical memory available

Question 32

Correct

3.00 points out of 3.00

What are the three states that a process might be in using a typical operating system?

Select one or more:

- a. Running ✓
- b. Blocked ✓
- c. User-mode
- d. Ready ✓
- e. Direct Execution
- f. Kernel-mode
- g. Swapped-out
- h. Mediated Execution

Your answer is correct.

Running ... currently executing.

Ready ... waiting to run.

Blocked ... unable to execute until some operation has completed or resource becomes available.

Discussed in Arpaci-Dusseau chapter 28.13

The correct answers are: Running, Blocked, Ready



Which of the following pairs of TLB features handle the same problems?

Select one or more:

- a. A fully associative TLB and a software managed TLB ✘ These are orthogonal. "Fully associative" refers to where particular translation entries can be stored in a TLB. "Software managed" refers to whether hardware or software is used to manage the TLB's contents.
- b. The TLB valid bit and the page table valid bit ✘ Page valid bit indicates if page has been allocated to the process. TLB valid bit indicates if TLB entry translation is valid.
- c. Flushing the TLB on context switches and using an address space identifier in TLB entries
- d. System call traps and TLB miss traps
- e. The TLB protection bits and the page table protection bits ✔ Both specify how a process is permitted to use a memory page (read-only, read/write, etc.)

Your answer is incorrect.

The correct answers are: Flushing the TLB on context switches and using an address space identifier in TLB entries, The TLB protection bits and the page table protection bits

[◀ Practice Midterm Exam](#)

Jump to...

[Lecture 10: Deadlocks - ... ▶](#)