CS/CE 232/324-L4 Operating Systems – Fall 2023

Instructor: Dr. Munzir Zafar

Date: September 14, 2023

10 minutes **Quiz 03** **Max Points: 100**

For each of the following questions, select only one correct answer from the multiple choices provided:

[Question 01] I/O communication can be handled using polling, or interrupt, or a hybrid where you poll a little while and then, if the device is not finished yet, use interrupt. Which of the following schemes will result in best performance:

	Slow I/O device	**Fast I/O device**	**I/O device with unknown speed**
a)	Polling	Interrupt	Hybrid
b)	Interrupt	Polling	Hybrid
c)	Polling	Polling	Polling
d)	Interrupt	Interrupt	Interrupt
e)	Hybrid	Hybrid	Hybrid

[Question 02] A Direct Memory Access (DMA) engine does NOT:

a) Help the CPU in the transfer of data between devices and main memory
b) Raise an interrupt when it completes its job
c) Result in improvement in performance by efficient handling of data transfer
d) Wait for instructions from the OS before doing a data transfer
e) Prevent the CPU from doing any other task during a data transfer

[Question 03] In hard disk drives, track-skewing means:

a) Circular tracks are slightly skewed into shapes of eclipse to allow faster jump to the neighboring track
b) Sectors in the neighboring track are ordered in a manner that helps with sequential read across neighboring tracks
c) Arm above the tracks is skewed at the end of the track for faster jump to neighboring tracks
d) Data transferred to/from a track is slightly re-arranged in order to accommodate the jump to a neighboring track
e) More sectors are accommodated at the outer tracks to allow packing more data on the tracks

[Question 04] For a disk with an RPM of 10,000, average seek time of 5 ms, and maximum transfer rate is 100 MB/s, the average rate of I/O for a random workload of 4KB would be:

a) 0.36 MB/s b) 0.5 MB/s c) 0.8 MB/s d) 50 MB/s e) 100 MB/s

[Question 05] The SCAN algorithm for disk scheduling is servicing a frozen queue of requests during a sweep when a new request comes for a track that has already been serviced in the current sweep. What happens?

a) It is handled immediately (i.e. not queued until the next sweep), if seeking the track will take shorter than servicing the next request in the on-going sweep

b) It is handled immediately (i.e. not queued until the next sweep), regardless of whether seeking the track will take longer or shorter than servicing the next request in the on-going sweep

c) It is not handled immediately (i.e. queued until the next sweep), if accessing the track will take shorter than servicing the next request in the on-going sweep

d) It is not handled immediately (i.e. queued until the next sweep), regardless of whether it will take longer or shorter than servicing the next request in the on-going sweep

e) It is handled immediately (i.e. not queued until the next sweep), if total positioning time to the requested block will take shorter than servicing the next request in the on-going sweep

[Question 06] Calling the `open()` function to open a file returns a file descriptor. Which of the following is true about this file descriptor:

a) Opening the same file will always return the same file descriptor, in any process

b) Opening the same file will always return the same file descriptor, in the current process

c) The value of the file descriptor depends on the order in which this file was opened by the current process (i.e. file descriptor can inform us if this was the first or the second file that was opened by this process)

d) The value of the file descriptor depends on the order in which this file was opened by any process (i.e. file descriptor can inform us if this was the first or the second file that was opened across the entire system)

e) The value of the file descriptor depends on the number of times this file was opened by the current process (i.e. file descriptor can inform us if this file is opened ten or twenty times by the current process)

f) The value of the file descriptor depends on the number of times this file was opened by the any process (i.e. file descriptor can inform us if this file is opened ten or twenty times across the entire system)

[Question 07] The same file is opened twice by the current process that returned two handles `fd1` and `fd2`. A `read()` for 100 bytes is called twice on both `fd1` and `fd2`. Then an `lseek(fd1, 100, SEEK_CUR)` is called. Then:

	Current offset of fd1	Current offset of fd2
a)	100	100
b)	100	200
c)	300	100
d)	300	200
e)	300	300

[Question 08] A file was opened by a parent process that returned a file descriptor. After that it created a child process using a `fork()` call. Then:

a) The file descriptor variables are local to each process but they point to the same entries in the open file table

b) The file descriptor variables are shared by both processes and they point to the same entries in the open file

c) The file descriptor variables are local to each process and they point to different entries in the open file table

d) The file descriptor variables are shared by both processes yet they point to different entries in the open file table

[Question 09] There was some dirty (i.e. not yet written) data in the write buffer when `fsync()` is called. This results in:

a) Forcing all dirty data to disk for all the files

b) Forcing all dirty data to disk for the file descriptor on which `fsync()` is called

c) Forcing the dirty data to disk only of the last write for the the file descriptor on which `fsync()` is called

d) Forcing the dirty data to disk for all writes to this file across the system, and not just the current process

e) Forcing the dirty data to disk for all writes to all files across the system, and not just the current process

[Question 10] The file foo has two hard links foo1, foo2, and one soft link foo3. When `rm foo` is called. It results in:

a) Deletion of the underlying file. So foo1, foo2, foo3, all links are broken and don't point to any file

b) Unlinking of foo only. So foo1, foo2, foo3, all still point to an underlying file

c) Unlinking of foo only. So foo1, foo2 still point to an underlying file, but foo3 no longer works

d) Deletion of foo, foo1, foo2 because they were hard links. But foo3 is not deleted because it was a soft link

e) Deletion of foo and foo3 because it was a soft link. But foo2, foo3 are intact because they were hard links.

CS/CE 232/324-L4 Operating Systems – Fall 2023

Instructor: Dr. Munzir Zafar

Date: 14th September, 2023

10 minutes **Quiz 03** **Max Points: 100**

For each of the following questions, select only one correct answer from the multiple choices provided:

[Question 01] I/O communication can be handled using polling, or interrupt, or a hybrid where you poll a little while and then, if the device is not finished yet, use interrupt. Which of the following schemes will result in best performance:

	Slow I/O device	Fast I/O device	I/O device with unknown speed
a)	Interrupt	Polling	Hybrid
b)	Polling	Polling	Polling
c)	Interrupt	Interrupt	Interrupt
d)	Hybrid	Hybrid	Hybrid
e)	Polling	Interrupt	Hybrid

[Question 02] A Direct Memory Access (DMA) engine does NOT:

a) Raise an interrupt when it completes its job
b) Result in improvement in performance by efficient handling of data transfer
c) Wait for instructions from the OS before doing a data transfer
d) Prevent the CPU from doing any other task during a data transfer
e) Help the CPU in the transfer of data between devices and main memory

[Question 03] In hard disk drives, track-skewing means:

a) Sectors in the neighboring track are ordered in a manner that helps with sequential read across neighboring tracks
b) Arm above the tracks is skewed at the end of the track for faster jump to neighboring tracks
c) Data transferred to/from a track is slightly re-arranged in order to accommodate the jump to a neighboring track
d) More sectors are accommodated at the outer tracks to allow packing more data on the tracks
e) Circular tracks are slightly skewed into shapes of eclipse to allow faster jump to the neighboring track

[Question 04] For a disk with an RPM of 10,000, average seek time of 5 ms, and maximum transfer rate is 100 MB/s, the average rate of I/O for a random workload of 4KB would be:

a) 0.5 MB/s b) 0.8 MB/s c) 50 MB/s d) 100 MB/s e) 0.36 MB/s

[Question 05] The SCAN algorithm for disk scheduling is servicing a frozen queue of requests during a sweep when a new request comes for a track that has already been serviced in the current sweep. What happens?

a) It is handled immediately (i.e. not queued until the next sweep), regardless of whether seeking the track will take longer or shorter than servicing the next request in the on-going sweep

b) It is not handled immediately (i.e. queued until the next sweep), if accessing the track will take shorter than servicing the next request in the on-going sweep

c) It is not handled immediately (i.e. queued until the next sweep), regardless of whether it will take longer or shorter than servicing the next request in the on-going sweep

d) It is handled immediately (i.e. not queued until the next sweep), if total positioning time to the requested block will take shorter than servicing the next request in the on-going sweep
e) It is handled immediately (i.e. not queued until the next sweep), if seeking the track will take shorter than servicing the next request in the on-going sweep

[Question 06] Calling the `open()` function to open a file returns a file descriptor. Which of the following is true about this file descriptor:
a) The value of the file descriptor depends on the order in which this file was opened by the current process (i.e. file descriptor can inform us if this was the first or the second file that was opened by this process)
b) The value of the file descriptor depends on the order in which this file was opened by any process (i.e. file descriptor can inform us if this was the first or the second file that was opened across the entire system)
c) The value of the file descriptor depends on the number of times this file was opened by the current process (i.e. file descriptor can inform us if this file is opened ten or twenty times by the current process)
d) The value of the file descriptor depends on the number of times this file was opened by the any process (i.e. file descriptor can inform us if this file is opened ten or twenty times across the entire system)
e) Opening the same file will always return the same file descriptor, in any process
f) Opening the same file will always return the same file descriptor, in the current process

[Question 07] The same file is opened twice by the current process that returned two handles `fd1` and `fd2`. A `read()` for 100 bytes is called twice on both `fd1` and `fd2`. Then an `lseek(fd1, 100, SEEK_CUR)` is called. Then:

	Current offset of fd1	Current offset of fd2
a)	100	200
b)	300	100
c)	300	200
d)	300	300
e)	100	100

[Question 08] A file was opened by a parent process that returned a file descriptor. After that it created a child process using a `fork()` call. Then:
a) The file descriptor variables are shared by both processes and they point to the same entries in the open file
b) The file descriptor variables are local to each process and they point to different entries in the open file table
c) The file descriptor variables are shared by both processes yet they point to different entries in the open file table
d) The file descriptor variables are local to each process but they point to the same entries in the open file table

[Question 09] There was some dirty (i.e. not yet written) data in the write buffer when `fsync()` is called. This results in:
a) Forcing all dirty data to disk for the file descriptor on which `fsync()` is called
b) Forcing the dirty data to disk only of the last write for the the file descriptor on which `fsync()` is called
c) Forcing the dirty data to disk for all writes to this file across the system, and not just the current process
d) Forcing the dirty data to disk for all writes to all files across the system, and not just the current process
e) Forcing all dirty data to disk for all the files in the current process

[Question 10] The file foo has two hard links foo1, foo2, and one soft link foo3. When `rm foo` is called. It results in:
a) Unlinking of foo only. So foo1, foo2, foo3, all still point to an underlying file
b) Unlinking of foo only. So foo1, foo2 still point to an underlying file, but foo3 no longer works
c) Deletion of foo, foo1, foo2 because they were hard links. But foo3 is not deleted because it was a soft link
d) Deletion of foo and foo3 because it was a soft link. But foo2, foo3 are intact because they were hard links.
e) Deletion of the underlying file. So foo1, foo2, foo3, all links are broken and don't point to any file

CS/CE 232/324-L4 Operating Systems – Fall 2023

Instructor: Dr. Munzir Zafar

Date: Sep 14, 2023

10 minutes **Quiz 03** **Max Points: 100**

For each of the following questions, select only one correct answer from the multiple choices provided:

[Question 01] I/O communication can be handled using polling, or interrupt, or a hybrid where you poll a little while and then, if the device is not finished yet, use interrupt. Which of the following schemes will result in best performance:

	Slow I/O device	**Fast I/O device**	**I/O device with unknown speed**
a)	Polling	Polling	Polling
b)	Interrupt	Interrupt	Interrupt
c)	Hybrid	Hybrid	Hybrid
d)	Polling	Interrupt	Hybrid
e)	Interrupt	Polling	Hybrid

[Question 02] A Direct Memory Access (DMA) engine does NOT:

a) Result in improvement in performance by efficient handling of data transfer
b) Wait for instructions from the OS before doing a data transfer
c) Prevent the CPU from doing any other task during a data transfer
d) Help the CPU in the transfer of data between devices and main memory
e) Raise an interrupt when it completes its job

[Question 03] In hard disk drives, track-skewing means:

a) Arm above the tracks is skewed at the end of the track for faster jump to neighboring tracks
b) Data transferred to/from a track is slightly re-arranged in order to accommodate the jump to a neighboring track
c) More sectors are accommodated at the outer tracks to allow packing more data on the tracks
d) Circular tracks are slightly skewed into shapes of eclipse to allow faster jump to the neighboring track
e) Sectors in the neighboring track are ordered in a manner that helps with sequential read across neighboring tracks

[Question 04] For a disk with an RPM of 10,000, average seek time of 5 ms, and maximum transfer rate is 100 MB/s, the average rate of I/O for a random workload of 4KB would be:

a) 0.8 MB/s b) 50 MB/s c) 100 MB/s d) 0.36 MB/s e) 0.5 MB/s

[Question 05] The SCAN algorithm for disk scheduling is servicing a frozen queue of requests during a sweep when a new request comes for a track that has already been serviced in the current sweep. What happens?

a) It is not handled immediately (i.e. queued until the next sweep), if accessing the track will take shorter than servicing the next request in the on-going sweep

b) It is not handled immediately (i.e. queued until the next sweep), regardless of whether it will take longer or shorter than servicing the next request in the on-going sweep

c) It is handled immediately (i.e. not queued until the next sweep), if total positioning time to the requested block will take shorter than servicing the next request in the on-going sweep

d) It is handled immediately (i.e. not queued until the next sweep), if seeking the track will take shorter than servicing the next request in the on-going sweep
e) It is handled immediately (i.e. not queued until the next sweep), regardless of whether seeking the track will take longer or shorter than servicing the next request in the on-going sweep

[Question 06] Calling the `open()` function to open a file returns a file descriptor. Which of the following is true about this file descriptor:
a) The value of the file descriptor depends on the number of times this file was opened by the current process (i.e. file descriptor can inform us if this file is opened ten or twenty times by the current process)
b) The value of the file descriptor depends on the number of times this file was opened by the any process (i.e. file descriptor can inform us if this file is opened ten or twenty times across the entire system)
c) Opening the same file will always return the same file descriptor, in any process
d) Opening the same file will always return the same file descriptor, in the current process
e) The value of the file descriptor depends on the order in which this file was opened by the current process (i.e. file descriptor can inform us if this was the first or the second file that was opened by this process)
f) The value of the file descriptor depends on the order in which this file was opened by any process (i.e. file descriptor can inform us if this was the first or the second file that was opened across the entire system)

[Question 07] The same file is opened twice by the current process that returned two handles `fd1` and `fd2`. A `read()` for 100 bytes is called twice on both `fd1` and `fd2`. Then an `lseek(fd1, 100, SEEK_CUR)` is called. Then:

	Current offset of fd1	Current offset of fd2
a)	300	100
b)	300	200
c)	300	300
d)	100	100
e)	100	200

[Question 08] A file was opened by a parent process that returned a file descriptor. After that it created a child process using a `fork()` call. Then:
a) The file descriptor variables are local to each process and they point to different entries in the open file table
b) The file descriptor variables are shared by both processes yet they point to different entries in the open file table
c) The file descriptor variables are local to each process but they point to the same entries in the open file table
d) The file descriptor variables are shared by both processes and they point to the same entries in the open file

[Question 09] There was some dirty (i.e. not yet written) data in the write buffer when `fsync()` is called. This results in:
a) Forcing the dirty data to disk only of the last write for the the file descriptor on which `fsync()` is called
b) Forcing the dirty data to disk for all writes to this file across the system, and not just the current process
c) Forcing the dirty data to disk for all writes to all files across the system, and not just the current process
d) Forcing all dirty data to disk for all the files in the current process
e) Forcing all dirty data to disk for the file descriptor on which `fsync()` is called

[Question 10] The file foo has two hard links foo1, foo2, and one soft link foo3. When `rm foo` is called. It results in:
a) Unlinking of foo only. So foo1, foo2 still point to an underlying file, but foo3 no longer works
b) Deletion of foo, foo1, foo2 because they were hard links. But foo3 is not deleted because it was a soft link
c) Deletion of foo and foo3 because it was a soft link. But foo2, foo3 are intact because they were hard links.
d) Deletion of the underlying file. So foo1, foo2, foo3, all links are broken and don't point to any file
e) Unlinking of foo only. So foo1, foo2, foo3, all still point to an underlying file