



CS609-System Programming  
(Solved Macq's)

**For Final Term**

**TOPIC 93 to 190**

**Also 70 to 90**



[Junaidfazal08@gmail.com](mailto:Junaidfazal08@gmail.com)  
[Bc190202640@vu.edu.pk](mailto:Bc190202640@vu.edu.pk)

FOR MORE VISIT  
VULMSHELP.COME

JUNAID MALIK  
0304-1659294



# AL-JUNAID TECH INSTITUTE

**PAID SERVICE  
CS619 PROJECTS**

Available training courses

- HTML
- JQUERY
- PHPMYSQL
- JAVASCRIPT
- BOOTSTRAPS
- NODE.JS
- REACT.JS
- CSS

**LMS HANDLING**

**PAID  
ASSIGNMENTS , QUIZ & GDB**

**95% RESULTS  
ALL LMS ACTIVITIES**



**Contact Us :**

**+92 304 1659294**

**[www.vulmshelp.com](http://www.vulmshelp.com)**

**[junaidfazal08@gmail.com](mailto:junaidfazal08@gmail.com)**

# AL-JUNAID TECH INSTITUTE

1. The GetCommandLine() function returns the command line as a single \_\_\_\_\_.
  - Float variable
  - Integer Array
  - Integer Variable
  - **Character string** **Topic 101**
2. Simple job management shell will allow \_\_\_\_\_ commands to run.
  - Two
  - Five
  - **Three** **Topic 104 (jobbg,jobs,kill)**
  - Four
3. In job management shell the shell uses \_\_\_\_\_ specific file keeping track of process ID and other related information
  - Shell
  - Process
  - System
  - **User** **Topic 104**
4. Process identify can be obtained from the \_\_\_\_\_ structure.
  - CreateProcess()
  - CREATE\_PROCESS()
  - **PROCESS\_INFORMATION** **Topic 93**
  - GetProcessInfo()
5. If the system is \_\_\_\_\_ time is multiplexed among multiple processes in an interleaved manner.
  - **Uniprocessor system** **Topic 99**
  - Linear processor system
  - Multicore processor system
  - Multiprocessor system
6. If a system is \_\_\_\_\_ then windows scheduler can run process threads on separate processors.
  - Uni processor
  - Linear processor
  - Single core processor
  - **Multiprocessor** **Topic 99**
7. FILETIME parameter of GetThreadTime() function is a \_\_\_\_\_ bit value.
  - 32
  - 8
  - 16
  - **64** **Topic100**
8. Closing the child process handle \_\_\_\_\_ the process.

# AL-JUNAID TECH INSTITUTE

- Does not create
  - Create
  - Destroy
  - Does not destroy Topic 93
9. Closing the child process only closes the access of the \_\_\_\_\_.
- All parent process
  - Parent process Topic 93
  - All children processes of parent process
  - Child process
10. In windows, there are \_\_\_\_\_ ways to get command line parameter for a process.
- Two Topic 96
  - Four
  - Five
  - Three
11. Ip ApplicationName handle's value \_\_\_\_\_ be NULL.
- Should
  - Should not According to me
  - May
  - May not
12. Control of a running fiber can be given to another fiber by using ----- function.
- GetCurrentFiber()
  - SwitchToFiber() Topic 129
  - CreateFiber()
  - ConvertThreadToFiberEx()
13. A thread can enable fiber operation by calling \_\_\_\_\_ function.
- GetCurrentFiber()
  - SwitchToFiber()
  - CreateFiber()
  - ConvertThreadToFiberEx() Topic 129
14. Inherited handles are \_\_\_\_\_ copies that a parent and child might be accessing.
- Connected
  - Similar
  - Distinct Topic 118
  - Related
15. A fiber can obtain its identity by calling \_\_\_\_\_ function.
- GetFiberData()
  - GetCurrentFiber() Topic 129
  - GetFiberId()
  - GetFiberIdentify()
16. The \_\_\_\_\_ function is used to obtain the process handle using the process ID.

# AL-JUNAID TECH INSTITUTE

- openProcesses()
  - GetCurrentProcess()
  - **OpenProcess()** Topic 93
  - OpenprocessID()
17. Parent process usually creates a \_\_\_\_\_ handle if parent and child process require \_\_\_\_\_ access rights.
- Duplication, same
  - **Duplication, different** Topic 94
  - New, same
  - New, different
18. There are ----- APIs to manage fiber.
- 10
  - 9
  - **7** Topic 129
  - 8
19. One process can finish another process using the function \_\_\_\_\_
- FinishProcess()
  - CreateProcess()
  - ExistProcess()
  - **TerminateProcess()** Topic 95
20. The simplest form of synchronization can be achieved through \_\_\_\_\_ construct.
- Run
  - Halt
  - Lock
  - **Wait** Topic 96
21. The single object for which the process waits is activated by \_\_\_\_\_.
- IpHandle
  - bWaitAll
  - **hHandle** Topic 96
  - nCount
22. The array of handles for which the process wait is activated by \_\_\_\_\_
- **IpHandle** Topic 96
  - hHandle
  - nCount
  - bWaitAll
23. \_\_\_\_\_ is the number of objects in an array. Should not exceed MAXIMUM WAIT OBJECTS
- IpHandle
  - hHandle
  - **nCount** Topic 96

# AL-JUNAID TECH INSTITUTE

- bWaitAll
24. \_\_\_\_\_ is the timeout period for wait. 0 for no wait and INFINITE for indefinitewait.
- IpHandle
  - hHandle
  - nCount
  - dwMilliseconds Topic 96
25. \_\_\_\_\_ handle describes if it's necessary to wait for all the objects to get free
- IpHandle
  - hHandle
  - nCount
  - bWaitAll Topic 96
26. How many The possible return values are \_\_\_\_\_
- 2
  - 3
  - 4
  - 5 Topic 95
27. The timeout period for wait() function is measured in \_\_\_\_\_.
- Millisecond Topic 96
  - Kilo hertz
  - Mega hertz
  - Microsecond
28. Its not a good idea to use \_\_\_\_\_ within the program as it will not give it a chance to release resources.
- createProcess()
  - TerminateProcess()
  - ExitProcess() Topic 95
  - FinishProcess()
29. A new fiber can be created by using \_\_\_\_\_
- ConvertThreadToFiberEx()
  - ConvertThreadToFiber()
  - CreateFiber() Topic 129
  - GetFiber()
30. In Environment Block, IpName is a \_\_\_\_\_ name.
- Stack
  - Array
  - Process
  - Variable Topic 97
31. In Environment Block, GetEnvironmentVariable() function returns \_\_\_\_\_ in case of failure.
- 0 Topic 97

# AL-JUNAID TECH INSTITUTE

- NULL
  - -1
  - UNDEFINED
32. In Environment Block, PATH is an example of an environment \_\_\_\_\_
- Stack
  - Array
  - Process
  - String Topic 97
33. An Environment Block is associated with \_\_\_\_\_ process(es) in the system.
- Some
  - No
  - Each and every Topic 97
  - Exactly one
34. In Environment Block (EB), each string is \_\_\_\_\_
- Five character long
  - Null terminated Topic 97
  - Undefined
  - Empty
35. The wait() function is limited to \_\_\_\_\_ handles.
- 32
  - 8
  - 64 Topic 98
  - 16
36. A \_\_\_\_\_ value will cause a thread to move from the running state to the ready state.
- Negative
  - INFINITE
  - 1
  - 0 Topic 127
37. Fiber which is shared to different threads \_\_\_\_\_
- Should not access private data
  - Should not access global data
  - Should access thread specific data
  - Should not access thread specific data Topic 128
38. Fibers are scheduled by the \_\_\_\_\_
- Hardware
  - Operating system
  - Application Topic 128
  - BIOS
39. IPC stands for
- Inter process communication My Point of View

# AL-JUNAID TECH INSTITUTE

- Inter privacy communication
  - Information process communication
  - Information and privacy communication
40. The sleep function allows a thread to move the \_\_\_\_ to the \_\_\_\_ state.
- Running, Suspended
  - Running, terminated
  - Running, ready
  - **Running,wait** **Topic 127**
41. The function SwitchToThread() provides a way for a thread to yield its processor to another \_\_\_\_ thread if there is one that is \_\_\_\_ to run.
- Ready, Running
  - Running, Running
  - **Ready, Ready** **Topic 127**
  - Running, Ready
42. The time period in sleep function is specified in \_\_\_\_.
- Seconds
  - **Milliseconds** **My point of View**
  - Nanoseconds
  - Microseconds
43. Default stack size for a thread is \_\_\_\_.
- **1MB** **Topic 126**
  - 1 Byte
  - 1 Bit
  - 1KB
44. A scheduler will move a \_\_\_\_ threads to the \_\_\_\_ state if the threads time slice expires without the thread waiting.
- Waiting, Waiting
  - **Running, Ready** **Topic 125**
  - Running, Waiting
  - Waiting, Ready
45. When a thread is suspended it goes into \_\_\_\_.
- Terminated state
  - Waiting state
  - Suspended state
  - **Ready state** **Topic 125**
46. A thread that is ready but do not have required resource will go into \_\_\_\_.
- Running state
  - Suspended state
  - **Waiting state** **My point of View**
  - Execution state

# AL-JUNAID TECH INSTITUTE

47. The scheduler will run the \_\_\_\_\_ priority \_\_\_\_\_ thread when a processor becomes available.

- Highest, ready Topic 125
- Lowest, ready
- Highest, running
- Lowest, running

48. If there are no `THREAD_PRIORITY_TIME_CRITICAL` thread, the process will run \_\_\_\_\_ first.

- `THREAD_PRIORITY_LOWEST`
- `THREAD_PRIORITY_NORMAL` My point of View
- `THREAD_PRIORITY_BELOW_NORMAL`
- `THREAD_PRIORITY_IDLE`

49. Which of the following priority is the highest class in the thread?

- `THREAD_PRIORITY_ABOVE_HIGHEST`
- `THREAD_PRIORITY_TIME_CRITICAL` My point of View
- `REALTIME_PRIORITY_CLASS`
- `THREAD_PRIORITY_HIGHEST`

50. The range of relative thread priorities is between \_\_\_\_\_.

- 0 to 1
- 0 to 4
- -2 to +2 (+2) Topic 123
- -1 to +1

51. Which of the following thread priorities will run first?

- Thread with `THREAD_PRIORITY_ABOVE_NORMAL` My point of View
- Thread with `THREAD_PRIORITY_BELOW_NORMAL`
- Thread with `THREAD_PRIORITY_IDEAL`
- Thread with `THREAD_PRIORITY_NORMAL`

52. In windows, most common processes have \_\_\_\_\_ priority class

- `HIGH_PRIORITY_CLASS`
- `IDLE_PRIORITY_CLASS`
- `NORMAL_PRIORITY_CLASS` My Point View
- `REALTIME_PRIORITY_CLASS`

53. The value returned using `TlsSetValue()` function is in the form of \_\_\_\_\_

- `BOOL` My Point of View
- `INT`
- `LPVOID`
- `DWORD`

54. In Thread Local Storage (TLS) arrangement, row represents \_\_\_\_\_ while column represent \_\_\_\_\_

- Thread TLS Index Topic 123

# AL-JUNAID TECH INSTITUTE

- Process Number, TLS Index not sure
  - TLS index, Thread
  - Thread, process Number
55. Which of the following function frees the specified index numbers?
- TlsClear()
  - **TlsFree()** Topic 123
  - TlsReSet()
  - TlsDelete()
56. API is used to allocate the index and it returns the TLS index in the form of the double word
- TlsClear()
  - **TlsAlloc()** Topic 123
  - TlsReSet()
  - TlsDelete()
57. \_\_\_\_\_ Provided valid indexes are used, The programmer can access TLS space using these simple GET/SET APIs
- TlsGetValue ()
  - **Both A and C** Topic 123
  - TlsSetValue ()
  - TlsDelete()
58. In the form of failure, TlsAlloc() function returns \_\_\_\_\_.
- DWORD
  - 0
  - **-1** Topic 123
  - BOOL
59. Which of the following is a correct statement?
- Every worker work with multiple processors on multiple processor
  - **Every worker work as a separate thread on a separate processor** Topic 122
  - Every worker work as a separate thread on a single processor
  - Every worker work with multiple processors on single processor
60. Which of the following is an optimal situation?
- After certain limit processor speed can also be enhanced
  - **After certain limit processor speed cannot be enhanced** Topic 122
  - Multiprocessing is not responsible for multiple flows of execution
  - Output of parallel program should not be same when it is serialized
61. Which of following statement is incorrect?
- **Program performance can be scaled without any certain limit.**
  - Program performance can be enhanced by using multithreading.
  - Program performance can be enhanced with Parallelism.
  - Program performance scales automatically, up to some limit.

# AL-JUNAID TECH INSTITUTE

62. Four threads are created i.e. thread 0, thread 1, thread 2, and thread 3 and are running to sort a large file, select the most appropriate statement?

- Wait for thread 1 to complete and merge it with thread 2.
- Wait for thread 0 to complete and merge it with thread 2.
- Wait for thread 1 to complete and merge it with thread 0. My point of view
- Wait for thread 2 to complete and merge it with thread 0.

63. Why a file is always mapped before accessing it?

- To access it just like accessing some data from main memory My point of view
- To make it secure
- To reuse it any time
- To access it in the form of LinkedList

64. Once all the threads are created, then can be run using \_\_\_\_\_ function.

- ResumeThread()
- ReadyThread()
- RunThread() Topic 121
- StartThread()

65. The basic difference between boss-worker thread model and client-server model is

- In boss worker all the thread are run at the boss's end, but in client-server model, each client run a different thread
- In the client-server all the thread are run at the server end, but in the boss worker model, each worker runs different thread Conceptual

66. Which of the following is not a thread model?

- Peer-to-peer Topic 119
- Pipeline
- Boss-worker
- Client-server

67. In client server model, rather than \_\_\_\_\_ work is done \_\_\_\_\_

- Concurrently, sequentially
- Problematic, Efficiency
- Sequentially, Concurrently Topic 119
- Linear, Straight

68. In which of the following models, work moves from one thread to the next thread?

- Peer-to-peer Model
- Pipeline Model Topic 119
- Boss-worker Model
- Client-server Model

69. The \_\_\_\_\_ flag is set to be \_\_\_\_\_ in the CreateProcess() function, which will determine whether child process will inherit copies of parent open handles.

- bInheritFlag, FALSE

# AL-JUNAID TECH INSTITUTE

- bInheritHandle, TRUE My point of View
  - bInheritFlag, TRUE
  - bInheritHandles, FALSE
70. Inherited handles are \_\_\_\_\_ copies that a parent and child might be accessing
- Similar
  - Connected
  - Distinct Wikipedia
  - Related
71. The get.JobNumber() function looks into the file for a vacant place. If no place is available, It appends a new record at the \_\_\_\_\_
- Next
  - Start
  - Middle
  - End My point of View
72. In getting a job number, the job Management Shell uses a number of job management functions to manage jobs information within the \_\_\_\_\_ life.
- Process
  - User Topic 103
  - Shell
  - System
73. Fiber management occurs at the \_\_\_\_\_ level.
- Hardware
  - User space Topic 128
  - Kernel
  - BIOS
74. In Asynchronous input/ Output \_\_\_\_\_.
- Input from the keyboard will be taken after the song is ended.
  - Song cannot be played until the other operations are not completed.
  - Song can be played with other operations being executed same time.
- Topic 119 also Conceptual (correct Accordingly)**
- Two operations cannot be run at the same time.
75. The first two parameters in argc will be \_\_\_\_\_ and \_\_\_\_\_ respectively.
- Process name, pattern which is to be searched Topic 118
  - Inputted file names, pattern to be searched
  - Pattern to be searched, process name
  - Inputted file names, process name
76. Total number of files inputted, can be obtained with \_\_\_\_\_ -
- Argc-1
  - Argc-2 Topic 118 (Confusion kindly Correct Accordingly)

# AL-JUNAID TECH INSTITUTE

- Argc-3
  - Argc
77. C Library Threading functions are \_\_\_\_\_ than windows library functions but not \_\_\_\_\_
- Simpler, Hard
  - Hard, Simpler
  - Diverse, Simpler
  - Simpler, Diverse **Topic 118**
78. LIBCMT is a \_\_\_\_\_
- BIOS Library for thread
  - Java Library for threads
  - C Library for threads **Topic 118**
  - Windows Library for threads
79. \_\_\_\_\_ function is used to extract \_\_\_\_\_ from a string.
- gettok(), Token
  - getTok, String
  - strtok(), String
  - strtok(), Token **Topic 118**
80. The return type of \_beingthreadex() function \_\_\_\_\_
- is HANDLE, but we need to type caste it for further processing.
  - Is not HANDLE, but we cannot type caste it for further processing.
  - is not HANDLE, we need to type caste it for further processing. **Topic 118 Conceptual**
  - is HANDLE, but we don't need to type caste for further processing.
81. \_\_\_\_\_ library function are thread safe than \_\_\_\_\_ library functions.
- C, java
  - Java, C
  - C, Windows **Topic 118**
  - Windows, C
82. Windows treats threads as \_\_\_\_\_.
- Objects **Topic 116**
  - Threads
  - Processes
  - Classes
83. When a thread exits, the thread \_\_\_\_\_ is deallocated and the handle referring to the thread in invalidated.
- Stack **Topic 113**
  - List
  - Queue
  - Array
84. ResumeThread() function will \_\_\_\_\_.

# AL-JUNAID TECH INSTITUTE

- Decrease the value of suspend count Topic 112
  - Reset the value of suspend count
  - Increase the value of suspend count
  - Not affect the value of suspend count
85. By default, the value of Suspend count, when creating a thread, is \_\_\_\_\_.
- 2
  - 1
  - 0 Topic 112
  - -1
86. Which of the following version of windows was not compatible with GetProcessIdOfThread() function?
- Windows 2003
  - Windows 7
  - Windows NT
  - Windows XP Topic 115
87. Which of the following function can be used to map a process with a thread?
- GetProcessIdOfThread() My Point of View
  - DWORD ResumeThread (HANDLE hThread)
  - DWORD SuspendThread (HANDLE hThread)
  - GetThreadOPendingFlag()
88. A thread will execute if and only if its Suspend Count value is \_\_\_\_\_
- Non zero
  - Above 1
  - 0 Topic 115
  - 1
89. Threads can also be treated as parent and child although the \_\_\_\_\_ is unaware of that.
- Kernel
  - Thread
  - Program
  - Operating system Topic 112
90. Threads share resources within a \_\_\_\_\_
- Code
  - Program
  - Thread
  - Process Topic 110
91. Threads uses the space assigned to a \_\_\_\_\_
- Thread
  - Process Topic 111
  - Code
  - Program

# AL-JUNAID TECH INSTITUTE

92. In a multi-threading system, multiple threads may exist within a single \_\_\_\_\_
- Thread
  - Process **Topic 109**
  - Code
  - Program
93. \_\_\_\_\_ is an independent unit of execution within a process.
- Thread **Topic 109**
  - Process
  - Code
  - Program
94. A job object is used to \_\_\_\_\_ process execution time and obtain user time statistics.
- Limit **Topic 108**
  - Write
  - Read
  - Open
95. The getJobNumber() function looks into the file for a vacant place. If no place is available, it appends a new recover of the file.
- Start
  - Next
  - Middle
  - End **My Point of View**
96. In job Management Shell, the shell uses \_\_\_\_\_ specific file keeping track of process ID and other related information.
- User **Topic 104**
  - Process
  - Shell
  - System
97. In finding a Process ID, the FindProcessId() obtains the process ID of the given job number. It simply looks on job number and \_\_\_\_\_ the record at the specific location.
- Sums
  - Executes
  - Reads **Topic 106**
  - Writes
98. Simple job management shell will allow \_\_\_\_\_ commands to run.
- Two
  - Three **Topic 104 (Jobbg, jobs, kill)**
  - Four
  - Five
99. In listing background jobs, the job management function uses display jobs () function. This function \_\_\_\_\_ a file

# AL-JUNAID TECH INSTITUTE

- Run
  - Start
  - Process
  - **Open** Topic 105
100. In Job Objects, firstly a job object is created using CreateJobObject(). It \_\_\_\_\_ a name and security attributes.
- Writes
  - Closes
  - **Uses** Topic 107
  - Throws
101. Control of a running fiber can be given to another fiber by using \_\_\_\_\_ function.
- GetCurrentFiber()
  - **SwitchToFiber()** Topic 129
  - CreateFiber()
  - ConvertThreadToFiberEx()
102. A thread can enable fiber operation by calling \_\_\_\_\_ function.
- GetCurrentFiber()
  - SwitchToFiber()
  - CreateFiber()
  - **ConvertThreadToFiberEx() OR ConvertThreadToFiber()** Topic 129
103. Fiber can obtain its identity by calling \_\_\_\_\_ function.
- GetFiberData()
  - **GetCurrentFiber()** Topic 129
  - GetFiberId()
  - GetFiberIdentity()
104. There are \_\_\_\_\_ APIs to manage fibers.
- 10
  - 9
  - **7** Topic 129
  - 8
105. A new fiber can be created by using \_\_\_\_\_
- ConvertThreadToFiberEx()
  - ConvertThreadToFiber()
  - **CreateFiber()** Topic 129
  - GetFiber()
106. The wait() function is limited to \_\_\_\_\_ handles
- 32
  - 16
  - 8
  - **64** Topic 98

# AL-JUNAID TECH INSTITUTE

107. Windows OS does not have any structure that keeps track record of the \_\_\_\_\_ processes.
- Child
  - Grand-child
  - Parent
  - **Parent-Child** **My point of View**
108. In comparison of DLL, executable library files are linked at \_\_\_\_\_ time.
- Link
  - Run
  - **Compiler** **My Point of View and GOOGLE**
  - Load
109. Interlocked functions are \_\_\_\_\_ to use.
- Simpler, Faster but hard
  - **Simpler, Faster and easy** **Topic 135**
  - Not simpler but faster and easy
  - Simpler easy but slow
110. Thread should not change the \_\_\_\_\_ environment.
- **Process** **Topic 137**
  - Hard disk
  - Integer
  - Ram
111. Which of the following function can be used to map the process with thread?
- DWORD Resume Thread (HANDLE hThread)
  - DWORD Suspend Thread (HANDLE hThread)
  - **GetProcessIdOfThread()** **My Point of View**
  - GetThreadIOPendingFlag()
112. Mutexes, Semaphores, Events, and Critical Section are the four synchronization objects provided by \_\_\_\_\_.
- Software
  - RAM
  - **Windows** **Topic 138**
  - processor
113. Like every other resource, threads are also treated as \_\_\_\_\_.
- Code
  - **Object** **Topic 112**
  - Program
  - Thread
114. The process Execution times uses the API GetCommandLine() to get the command line as a single \_\_\_\_\_.
- Bool

# AL-JUNAID TECH INSTITUTE

- Float
  - **String** Topic 102
  - None
115. lpApplicationName \_\_\_\_\_ be NULL.
- May
  - Should
  - **Should not** My point of View
  - May not
116. Volatile stage is a \_\_\_\_\_ level facility.
- **Compiler or windows** Topic 133
  - Programmer
  - BIOS
  - Hardware
117. CRITICAL\_SECTION Objects do not have \_\_\_\_\_.
- Loops
  - **Handler** Topic 139
  - Variables
  - Process
118. CRITICAL\_SECTION Objects do not have \_\_\_ and are not shared among the \_\_\_\_\_.
- Handler
  - **Both A and C** Topic 139
  - process
  - Process
119. \_\_\_\_\_ variables should not be accessible globally.
- String
  - **Locally required** Topic 139
  - Long
  - Integer
120. Thread IDs and handles can be obtained using functions quite similar to the ones used with \_\_\_\_\_
- **Process** Topic 114
  - Program
  - Thread
  - Object
121. A thread can be signaled from which of the following?
- ResumeThread()
  - CloseThread()
  - CreateThread()
  - **ExitThread() OR TerminateThread()** Topic 117

# AL-JUNAID TECH INSTITUTE

122. If there are 129 objects, how many times we will have to call the wait () function?
- 4
  - 6
  - 3 **My Point of View**
  - 2
123. ResumeThread() function will \_\_\_\_
- Increase
  - Decrease **Conceptual**
  - None of given
  - Both
124. If you want to create 4 thread i.e, thread 0, thread 1, thread 2, and thread 3, they all must be created at \_\_\_\_ state.
- Suspended **Topic 120**
  - Blocked
  - Running
  - Ready
125. It must be ensured that \_\_\_\_ thread(s) is/ are not modifying \_\_\_\_ data at same time.
- Many, Many
  - 1, Many
  - Many,1 **Topic 131 Conceptual**
  - 1,1
126. The result of concurrent processing and normal processing \_\_\_\_
- Yield same output
  - Is always different
  - Is always same
  - May differ in some cases **My point of View**
127. In memory architecture and barriers, before writing the value s back to memory, the processor usually keeps them in \_\_\_\_
- Hard disk
  - Memory card
  - Processor
  - Cache **Topic 134**
128. \_\_\_\_ are used to ensure that memory is accessed in the desired order.
- Memory barriers or memory fences **Topic 134**
  - Memory paths or memory houses
  - Multi processors
  - Memory blocks or memory chains
129. Turning off \_\_\_\_ may adversely affect the performance but sometimes may \_\_\_\_ the program.

# AL-JUNAID TECH INSTITUTE

- Usability, faster
  - Optimization, slow Topic 133
  - Performance, slow
  - Security, faster
130. Turning off Optimization may adversely affect the performance but sometimes may slow the program. ANSI C provides a qualifier \_\_\_\_\_ for this purpose.
- Volatile Topic 131
  - Non-volatile
  - Both A and B
  - None of Given
131. The volatile qualifier does not guarantee that the modifications will be visible to \_\_\_\_\_ in a desired order.
- Processors Topic 134
  - User
  - Ram
  - Hard disk
132. The \_\_\_\_\_ functions provide memory barriers.
- Interlocked Topic 139
  - User define
  - Interchange
  - Intermediate
133. Interlocked increment () takes a 32-bit signed variable as an arguments, which should be placed in memory at the \_\_\_\_\_ boundary.
- 4-byte Topic 135
  - 8-byte
  - 16-byte
  - 2-byte
134. If a variable with a volatile scope only needs to be incremented, decremented, or exchanged, interlocked functions are the \_\_\_\_\_ choice.
- Last
  - Best Topic 135
  - 2<sup>nd</sup> last
  - Worst
135. \_\_\_\_\_ are most suited if variable with volatile scope only need to be incremented,decremented and exchanges
- Interlocked functions Topic 135
  - Local
  - Global
  - None
136. The most basic of Interlocked Functions

# AL-JUNAID TECH INSTITUTE

- InterlockedIncrement()
  - InterlockedDecrement()
  - Both A and B Topic 135
  - None of Given
137. How many Basic Function of Interlocked()
- 1
  - 2 Topic 135
  - 3
  - 4
138. If any needed variables are not initialized, then create \_\_\_\_\_ threads until variables are initializes.
- Suspended Topic 137
  - Double
  - Slow
  - Fast
139. In thread safe code, \_\_\_\_\_ conditions are avoided.
- Race Topic 137
  - Loop
  - Local
  - Global
140. When more than one thread can run the same code without introducing synchronization issues, it is said to be \_\_\_\_\_
- Thread safe Topic 137
  - Local
  - Threaded
  - Initialized properly
141. InterlockedIncrement64() and interlockedDecrement64() can be utilized on 64-bit systems if the addend is paced at the \_\_\_\_\_ boundary.
- 8-byte Topic 135
  - 4-byte
  - 2-byte
  - 16-byte
142. \_\_\_\_\_ have a performance disadvantage since they create a memory barrier.
- Interchanged function
  - Interlocked function Topic 135
  - Intermediate function
  - Interlocked memories
143. If a \_\_\_\_\_ is used to store thread-specific data, other threads will also use it.
- Small variable
  - Large variable

# AL-JUNAID TECH INSTITUTE

- Global variable topic 136
- Local variable
- 144. Make a variable \_\_\_\_\_ if we know it includes thread-specific information.
- Static
- Local Topic 136
- Global
- Random
- 145. According to the criterion for good thread code, \_\_\_\_\_ storage should not be used for the purpose of local storage.
- Hard disk
- Global Topic 136
- Cache
- Ram
- 146. If we know the information in a variable will be used by other threads, we can make it \_\_\_\_\_.
- Global Topic 136
- Local
- Main
- Auto
- 147. When using thread synchronization objects, there are always inherent risks associated with them such as \_\_\_\_\_
- Lower power
- Multi clocks
- Load
- Deadlocks Topic 137
- 148. A thread waits for other threads to \_\_\_\_\_ using the waitForObject() or waitForMultipleObjects() methods.
- Wait
- Terminate Topic 138
- Start
- Update
- 149. The part of the code that can only be executed by one thread at a time is referred to as the \_\_\_\_\_
- Process section
- Loop section
- Top section
- Critical section Topic 139
- 150. Only \_\_\_\_\_ can be in the CRITICAL\_SECTION variable at a time
- Odd threads
- Many threads

# AL-JUNAID TECH INSTITUTE

- One thread Topic 139
- Even threads
151. Which function is used to initialize the CRITICAL\_SECTION.
- InitializeCriticalSection() Topic 139
- DeleteCriticalSection()
- Both A and B
- None of Given
152. Which function is used to Delete the CRITICAL\_SECTION.
- InitializeCriticalSection()
- DeleteCriticalSection() Topic 139
- Both A and B
- None of Given
153. Multiple threads may call EnterCriticalSection() but only \_\_\_\_\_ thread is allowed to enter the critical section while the rest are blocked.
- One Topic 139
- Two
- Three
- Four
154. A call to LeaveCriticalSection() must match a/ an \_\_\_\_\_
- EnterSection()
- Variables
- EnterCriticalSection() Topic 139
- CriticalSectionPart()
155. CRITICAL\_SECTION object do not have \_\_\_\_\_
- Process
- Variables
- Handles Topic 139
- Loops
156. EnterCriticalSection() can be called by \_\_\_\_\_ but only one of them is allowed to enter the critical section, while the others are blocked.
- Two threads
- One thread Topic 139
- Half thread
- Many threads
157. Using the critical section construct is \_\_\_\_\_ and intuitive.
- Easy Topic 140
- Difficult
- Short
- Long

# AL-JUNAID TECH INSTITUTE

158. It would be \_\_\_\_ to use different objects in the same thread or across numerous threads that share the same data.
- Easy
  - **Incorrect** Topic 141
  - Correct
  - Difficult
159. For mutual exclusion to work, \_\_\_\_ variables must be protected by a single object across all threads.
- Integer
  - Important
  - **Shared** Topic 141
  - specific
160. Within the critical section, all the variables must be guarded by \_\_\_\_
- Critical object
  - All objects
  - **A single object** Topic 141
  - Double object
161. What is the limit for object to be waited for waitForMultiple Objects () function in window?
- 63
  - 65
  - 62
  - **64** Topic 141
162. Producer consumer problem is a classical problem in \_\_\_\_
- Critical Section
  - Conical collection
  - **Mutual Exclusion** Topic 142
  - None of given
163. The producer periodically \_\_\_\_ a message.
- **Creates** Topic 142
  - Delete
  - Update
  - None
164. The producer also computes a simple \_\_\_\_ of the message
- **Checksum** Topic 142
  - Contents
  - Update
  - None
165. \_\_\_\_ is a short form of Mutual Exclusion
- Mechanism

# AL-JUNAID TECH INSTITUTE

- Checksum
  - **Mutex** Topic 143
166. Which of the following Windows functions used to manage mutexes.
- CreateMutex()
  - ReleaseMutex()
  - OpenMutex()
  - **All of the given** Topic 144
167. Home Many windows Function are used to manage mutex.
- 1
  - 2
  - **3** Topic 144
  - 4
168. lpMutexName is the name of the mutex.
- **True** Topic 144
  - False
169. OpenMutex() is used to \_\_\_\_\_ and \_\_\_\_\_ named mutex
- Existing,open
  - **Open,Existing** Topic 144
  - Open,Delete
  - Exsiting,Delete
170. However, abrupt termination of a thread indicates a serious programming flaw. Mutex waits can \_\_\_\_\_
- Time In
  - **Time Out** Topic 146
  - Has
  - Does Not
171. If NULL is returned it indicates dose not failure.
- True
  - **False** Topic 144
172. Semaphore maintains a count
- **True** Topic 147
  - False
173. A semaphore is in a signaled state when the count is greater than.
- 1
  - -1
  - **0** Topic 147
  - None of given
174. Semaphore is \_\_\_\_\_ when the count is 0.
- Signaled

# AL-JUNAID TECH INSTITUTE

- Un-sigaled Topic 147
- Release
- None of the given
175. Which of the following are used in semaphore Count?
- CreateSemaphore()
- CreateSemaphoreEx()
- OpenSemaphore()
- All of the Given Topic 147
176. cReleaseCount gives the count after the release and must be greater than \_\_\_\_.
- 1
- -1
- 0 Topic 147
- None of given
177. Events are classified as either
- manual-reset
- auto-reset
- Both A and B Topic 150
178. A \_\_\_\_\_ event can signal several waiting threads simultaneously and can be reset.
- manual-reset Topic 150
- auto-reset
179. An \_\_\_\_\_ event signals a single waiting thread, and the event is reset automatically.
- manual-reset
- auto-reset Topic 150
180. If bInitialState is TRUE, the event is set to a \_\_\_\_\_ state
- Signaled Topic 150
- Un-Sigaled
181. If the event is manual-reset, it remains signaled until a thread explicitly calls ResetEvent().
- True Topic 150
- False
182. How many ways to use Events?
- 1
- 2
- 3
- 4 Topic 151
183. Windows operating system is a multithreaded \_\_\_\_\_ that provides support for real time applications and multiprocessors.
- Hardware

# AL-JUNAID TECH INSTITUTE

- User space
  - Kernel Topic 153
  - BIOS
184. Program design and performance can be simplified and improved by
- Odd threads
  - Many threads
  - Threads Topic 153
  - Even threads
185. \_\_\_\_\_ Synchronization is a way to coordinate processes that use shared data
- Shell
  - Process Topic 153
  - System
  - User
186. Which of the following Synchronization objects.
- Critical Section
  - Semaphore
  - Mutex
  - All of the Given Topic 153
187. The \_\_\_\_\_ functions provide a simple mechanism for synchronizing access to a variable that is shared by multiple threads.
- Interlocked Topic 155
  - Local
  - Global
  - None
188. Memory allocation is then performed using using \_\_\_\_\_ rather than using \_\_\_\_\_
- HeapAlloc() and HeapFree(), malloc() and free() Topic 56
  - malloc() and free() and HeapAlloc() and HeapFree(),
189. Each thread that performs memory management can create a Handle to its own heap using \_\_\_\_\_
- HeapCreate() Topic 156
  - HeapAlloc()
  - HeapFree()
  - None of the given
190. Threads allocate memory and free memory using \_\_\_\_\_ functions respectively
- malloc () and free () Topic 156
  - free () and malloc ()
191. How may aspects compare the performance on the Basis.
- 1
  - 2

# AL-JUNAID TECH INSTITUTE

- 3 Topic 157
- 4
192. Which of the Following aspects that are used to compare the Performance.
- Real-Time
- User-Time
- System-time
- All of the given Topic 157
193. MX(Mutexes) version costs more than \_\_\_\_\_ times than IN
- 2 times
- 2 to 30 times Topic 158
- 1 to 30 times
- 3 times
194. CS (Critical Section) version costs \_\_\_\_\_ times more than IN
- 2 to 30 times
- 2 times Topic 158
- 1 to 30 times
- 3 times
195. For older version of windows CS was not scalable.
- True Topic 159
- False
196. Critical Section works in \_\_\_\_\_
- Hardware
- User space Topic 161
- Kernel
- BIOS
197. How many methods of Lightweight Reader Writer Locks.
- 1
- 2 Topic 163
- 3
- 4
198. Which of the following methods of Lightweight Reader Writer Locks
- Exclusive Mode
- Shared Mode
- Both A and B Topic 163
- None of given
199. How many APIs which are used to access SRWs in Exclusive mode.
- 1
- 2 Topic 164
- 3
- 4

# AL-JUNAID TECH INSTITUTE

200. When a thread is created almost \_\_\_\_\_space is reserved for that thread and with increasing threads memory area is piled up.

- 1 MB Topic 166
- 1 KB
- 1 Bit
- 1 Byte

201. If you \_\_\_\_\_ the threads the you can lose the benefits of parallelism and synchronization

- Minimize Topic 166
- Maximize

202. Which of the following optimization technique?

- Use of semaphore throttles
- Asynchronous I/O
- Using I/O completion ports
- All of the Given Topic 166

203. Every process has a dedicated \_\_\_\_\_ pool.

- Odd threads
- Many threads
- Threads Topic 172
- Even threads

204. The thread pool is used to \_\_\_\_\_the number of application threads and provide management of the worker threads

- Increased
- Reduce Topic 172
- Constant
- Equal

205. How many types of call back Function.

- 1
- 2 Topic 173
- 3
- 4

206. How many sorts of parallelism?

- 1
- 2 Topic 177
- 3
- 4

207. Which of the following parallelism define every loop iteration can execute concurrently.

- Loop parallelism Topic 177

# AL-JUNAID TECH INSTITUTE

- Fork-join parallelism
208. \_\_\_\_\_ the control flow divides (like the shape of the fork) into multiple flows that join later
- Loop parallelism
  - Fork-join parallelism Topic 177
209. Each process has its own process affinity mask and a system affinity mask.
- True Topic 180
  - False
210. How many Bits vector of MASK?
- 1
  - 2
  - 3 Topic 180
  - 4
211. Which of the following Bits vector of MASK?
- System Affinity Mask
  - Process Affinity Mask
  - Thread Affinity Mask
  - All of the Given Topic 180
212. IPC stand for \_\_\_\_\_
- Internet process control
  - Inter-process Communication Topic 180
  - Inter-parameter control
  - None of the Given
213. How many Types of Pipe?
- 1
  - 2 Topic 181 (Anonymous Pipes, Named Pipes)
  - 3
  - 4
214. File-like object called Pipe can be used for IPC.
- True Topic 181
  - False
215. Which of the following are correct statement of Anonymous Pipes?
- Simple anonymous pipes are character based and half duplex. Topic 181
  - They allow network wide communication.
216. Which of the following are correct statement of Named Pipes?
- There can be multiple open handles for a pipe
  - They allow network wide communication.
  - Much more powerful than anonymous pipes
  - All of the Given Topic 181

# AL-JUNAID TECH INSTITUTE

217. Anonymous pipes allow \_\_\_\_\_ (half-duplex) communication
- one-way Topic 182
  - two-way
218. The default set of cbPipe is \_\_\_\_\_
- 0 Topic 182
  - 1
  - 2
  - 3
219. Named Pipe is \_\_\_\_\_
- Directional
  - Bi-Directional Topic 184
  - Tri-Directional
  - None of the Given
220. The pipe name would be like this.
- `\\.pipe\[path]pipename` Topic 187
  - `\\servername\pipe\pipename`
221. Which of the following connection sequences for the Client.
- The client connects with a server
  - Communicates with the server using CreateFile()
  - Performs Read and Write operations and ultimately disconnects
  - All of the given Topic 190
222. Which of the following connection sequences for the Server.
- Communicates with the client.
  - As a result ReadFile() returns FALSE
  - The server-side connection is disconnected
  - All of the given Topic 190
223. Which of the following Returns information whether the pipe is in blocking or non-blocking mode, message oriented or byte oriented, number of pipe instances, and so on.
- SetNamedPipeHandleState()
  - GetNamedPipeHandleState() Topic 188
  - GetNamedPipeInfo()
  - None of the given
224. Which of the following Allows the program to set the same state attributes. Mode and other values are passed as reference so that NULL can also be passed indicating no change is desired.
- SetNamedPipeHandleState() Topic 188
  - GetNamedPipeHandleState()
  - GetNamedPipeInfo()
  - None of the given

# AL-JUNAID TECH INSTITUTE

225. Which of the following Determines whether the handle is for client or a server, buffer sizes, and so on.
- SetNamedPipeHandleState()
  - GetNamedPipeHandleState()
  - **GetNamedPipeInfo()** Topic 188
  - None of the given
226. Call \_\_\_\_\_ to disconnect from the handle
- **DisconnectNamedPipe()** Topic 189
  - WaitNamedPipe()
  - ConnectNamedPipe()
  - All of the given
227. \_\_\_\_\_ is used to synchronize connections to the server
- DisconnectNamedPipe()
  - **WaitNamedPipe()** Topic 189
  - ConnectNamedPipe()
  - All of the given
228. \_\_\_\_\_ are the security attributes as discussed previously
- **lpSecurityAttributes** Topic 186
  - nOutBufferSize
  - dwOpenMode
  - nMaxInstance
229. \_\_\_\_\_ and \_\_\_\_\_ give the size in bytes of input and output buffer
- **nOutBufferSize and nInBufferSize** Topic 186
  - nInBufferSize and nOutBufferSize
230. which of the following determines maximum number of pipe instances?
- lpSecurityAttributes
  - nOutBufferSize
  - dwOpenMode
  - **nMaxInstance** Topic 186
231. which of the following indicates whether writing is message oriented or byte oriented?
- **dwPipeMode** Topic 186
  - nOutBufferSize
  - dwOpenMode
  - nMaxInstance
232. \_\_\_\_\_ is the default timeout period.
- **nDefaultTimeOut** Topic 186
  - nOutBufferSize
  - dwOpenMode
  - nMaxInstance

# AL-JUNAID TECH INSTITUTE

233. The period (.) stands for local machine.
- True **Topic 186**
  - False
234. In mutex which type of data structure that stores the resource should also be used to store the mutexes because mutexes correspond to the resources.
- Different **Topic 154**
  - Same
  - Equal
  - None of the Given
235. Only the \_\_\_\_\_ Scheduler decides which thread has the priority according to its scheduling policy.
- Hardware
  - Application
  - OS **Topic 154**
  - BIOS
236. The shell uses a ..... file keeping track of process ID and other related information?
- User-specific **Topic 104**
237. The array of handles for which the process waits.
- lpHandle
238. In Listing Background Jobs the job management function used for the purpose is DisplayJobs(). The function \_\_\_\_\_ the file.
- Process
  - Runs
  - Opens **Topic 105**
  - Starts
239. In using Job objects, job objects are used to \_\_\_\_\_ process execution time and obtain user time statistics.
- Open
  - Limit **Topic 108**
  - Read
  - White
240. FILETIME is a(an) \_\_\_\_\_-bit value. GetThreadTime() can be used similarly and required a hread handle.
- 32
  - 64 **Topic 100**
  - 16
  - 8
241. In Finding a process Id, the FindProcessId() obtains the process Id of given job number. It simply looks up into the File based on the job number and \_\_\_\_\_ the record at the specific location.

# AL-JUNAID TECH INSTITUTE

- Executes
- Reads Topic 106
- Writes
- Sums

242. Thread Issues Threads Share Resources Within a \_\_\_\_\_. One Thread May inadvertently another Threads' Data.

- Process Topic 10
- Threads
- Program
- Code

243. An Environment Block is associated with \_\_\_\_\_ process.

- Some
- One
- Each Topic 97
- No

**From 70 to 90 TOPIC**

244. When a fixed size data structure is allocated from a single heap, it reduces \_\_\_\_\_

- ❖ Fragmentation
- ❖ Errors
- ❖ Memory density
- ❖ Throughput

245. The heapReAlloc() API has \_\_\_\_\_ parameter(s).

- ❖ 4
- ❖ 1
- ❖ 3
- ❖ 2

246. The heapAlloc() API has \_\_\_\_\_ parameter(s).

- ❖ 3
- ❖ 4
- ❖ 2
- ❖ 1

247. When a heap (logical structure) is created the memory is \_\_\_\_\_ allocated at the program.

- ❖ Partially
- ❖ Completely

# AL-JUNAID TECH INSTITUTE

❖ Not directly

❖ Directly

248. \_\_\_\_\_ are the APIs for heap memory allocation.

❖ Heapcreate ()and HeapRealloc()

❖ Allocheap () and HeapRealloc()

❖ HeapAlloc() and HeapRealloc()

❖ HeapAlloc() and HeapRealloc()

249. For a non growable heap, the value of dwbytes in heap memory allocation is

❖ 0\*7FEE8

❖ 0\*7FDD8

❖ 0\*AAAA8

❖ 0\*7FFF8

250. \_\_\_\_\_ is the first step to allocate heap in a program.

❖ HeapDestroy()

❖ HeapFree()

❖ Release and handle

❖ Get heap handle

251. The function heapSize() returns the size of a block, or \_\_\_\_\_ in case failure.

❖ NULL

❖ 1

❖ -1

❖ 0

252. \_\_\_\_\_ is used to deallocate the entire heap.

❖ HeapDestroy()

❖ HeapFree()

❖ HeapTruncate()

❖ HeapDelete()

253. Sorting is performed in the \_\_\_\_\_

❖ RootHeap

❖ RecHeap

❖ ProcHeap

❖ NodeHeap

254. \_\_\_\_\_ stores the root address.

❖ RootHeap

❖ RecHeap

❖ ProcHeap

# AL-JUNAID TECH INSTITUTE

- ❖ NodHeap
255. The NodeHeap maintains a \_\_\_\_\_
- ❖ Data
  - ❖ **Data structure**
  - ❖ Record
  - ❖ Root
256. There are \_\_\_\_\_ parameters taken by the HeapCreate() API.
- ❖ **3**
  - ❖ 4
  - ❖ 2
  - ❖ 1
257. Which of the following is the correct windows API for accessing heap?
- ❖ INT GetProcessHeap(VOID)
  - ❖ VOID GetProcessHeap(HANDLE)
  - ❖ **HANDLE GetProcessHeap(VOID)**
  - ❖ INT\*GetProcessHeap(VOID)
258. When a fixed size data structure is allocated from a single heap, it reduces \_\_\_\_\_
- ❖ Memory density
  - ❖ Errors
  - ❖ Throughput
  - ❖ **Fragmentation**
259. The parameters “flOption” in the HeapCreate() API is a combination of \_\_\_\_\_ flafs.
- ❖ 1
  - ❖ 2
  - ❖ 4
  - ❖ **3**
260. While using CreateFileMapping(), \_\_\_\_\_ allow the mapping object to be secured.
- ❖ INVALID\_VALUES
  - ❖ PSECURITY\_ATTRIBUTES
  - ❖ **LPSECURITY\_ATTRIBUTES**
  - ❖ INVALID\_HANDLE\_VALUES
261. While using CreateFileMapping(), setting lpMapName to \_\_\_\_\_ disables the map sharing.
- ❖ -1
  - ❖ **NULL**

# AL-JUNAID TECH INSTITUTE

- ❖ 0
  - ❖ 1
262. \_\_\_\_\_ is the API for file mapping objects.
- ❖ Create\_File\_Mapping()
  - ❖ **CreateFileMapping()**
  - ❖ FileCreateMapping()
  - ❖ MakeFileMapping()
263. Which of the following are the number of parameters taken by CreateFileMapping()?
- ❖ 7
  - ❖ **6**
  - ❖ 5
  - ❖ 4
264. The \_\_\_\_\_ -- flag is set to be \_\_\_\_\_ in the CreateProcess() function, which will determine whether child process will inherit copies of parent open handles.
- ❖ blnheritFlag, TRUE
  - ❖ blnheritHandles, FALSE
  - ❖ blnheritFlag, FALSE
  - ❖ **bInheritHandles, TRUE**
265. IPC stands for \_\_\_\_\_.
- ❖ Information and privacy communication
  - ❖ Inter privacy communication
  - ❖ Information process communication
  - ❖ **Inter Process Communication**
266. Inherited handles are \_\_\_\_\_ copies that a parent and child might be accessing.
- ❖ Connected
  - ❖ Similar
  - ❖ related
  - ❖ **Distinct**
267. Process IDs are always \_\_\_\_\_.
- ❖ Frequent
  - ❖ Repeated
  - ❖ Constant
  - ❖ **Unique**
268. The process obtains environment and other information from \_\_\_\_\_ call.
- ❖ CreateThread()
  - ❖ GetEnvironmentinfo()

# AL-JUNAID TECH INSTITUTE

- ❖ Getinfo()
  - ❖ **CreateProcess()**
269. lpApplicationName handle's value \_\_\_\_\_ be NULL.
- ❖ May not
  - ❖ May
  - ❖ should
  - ❖ **Should not**
270. In windows there are \_\_\_\_\_ ways to get command line parameters for a process.
- ❖ Five
  - ❖ Four
  - ❖ **Two**
  - ❖ Three
271. Windows OS does not have structure that keeps track record of the \_\_\_\_\_ processes.
- ❖ Child
  - ❖ Grand-child
  - ❖ Parent
  - ❖ **Parent\_Child**
272. The most fundamental process management function in windows is CreateProcess() that has \_\_\_\_\_ parameters.
- ❖ 12
  - ❖ 6
  - ❖ 4
  - ❖ **10**
273. The process can share memory and files but the process itself lie an individual \_\_\_\_\_ memory space .
- ❖ Non\_volatile
  - ❖ Physical
  - ❖ permanent
  - ❖ **Virtual**
274. Thread Local Storage (TLS) is an array of collection of pointers enabling a thread to \_\_\_\_\_ storage to create its unique data environment.
- ❖ De-allocate
  - ❖ Clear
  - ❖ Re-allocate
  - ❖ **Allocate**
275. Each thread has its own \_\_\_\_\_.

# AL-JUNAID TECH INSTITUTE

- ❖ TLS
- ❖ Environment Block
- ❖ Stack
- ❖ TLS and Stack

276. The process of DLL detachment in explicit linking is invoke by \_\_\_\_\_ function call.

- ❖ Free()
- ❖ freeLib()
- ❖ Flibra
- ❖ FreeLibrary()

277. Information regarding DLLs is placed in the \_\_\_\_\_ data structure.

- ❖ dwBuilderNumber
- ❖ dwPlatform
- ❖ MAJORVERSION
- ❖ DLLVERSION

278. LoadLibrary() and LoadLibraryEx() should never be called from \_\_\_\_\_ as it will create more DLL entry Points.

- ❖ ThreadLibrarycalls()
- ❖ DllMinFunc()
- ❖ DisableThreadLibraryCalls()
- ❖ DllMain()

279. LoadLibraryEx() can suppress the execution of entry point, in \_\_\_\_\_ -- linking of DLL.

- ❖ Implicit
- ❖ Static
- ❖ Dynamic
- ❖ Explicit

280. "Application that require newer updated functionality may sometime link with older DLL version". This statement refers to \_\_\_\_\_ of DLL versioning

- ❖ Strength
- ❖ Advantages
- ❖ Caution
- ❖ Problem

281. If entry point of DLL is not specified, then it is an example of \_\_\_\_\_ -- linking.\

- ❖ Explicit
- ❖ Dynamic
- ❖ Hard

# AL-JUNAID TECH INSTITUTE

❖ **Implicit**

282. In case of \_\_\_\_\_ linking the DLL attaches at the time of process start and detaches when process ends

- ❖ Explicit
- ❖ Dynamic
- ❖ Hard
- ❖ **Implicit**

283. Explicit linking requires the program to explicitly specify the DLL to be \_\_\_\_\_.

- ❖ Freed
- ❖ Loaded
- ❖ Loaded and freed

❖ **Ans: Loaded or freed**

284. In a pointer function declaration for DLL explicit linking, HMODULE is NULL in case of \_\_\_\_\_.

- ❖ Execution
- ❖ Waiting
- ❖ success

❖ **Failure**

285. Once the DLL is loaded, the programmer needs to obtain \_\_\_\_\_ into the DLL for an entry point.

- ❖ Dynamic address
- ❖ Physical address
- ❖ Bus address

❖ **Procedure Address**

286. We write and \_\_\_\_\_ function in DLL and invoke them explicitly

- ❖ Compile
- ❖ Encrypt
- ❖ decrypt

❖ **Encapsulate**

287. In DLLs the executable library files are linked at \_\_\_\_\_ time

❖ **. Ans: Compile**

288. Each DLL program will have its own copy of \_\_\_\_\_ variables.

❖ **Ans: Globle**

289. In \_\_\_\_\_ operating system DLLs are used to invoke all kernel services.

❖ **Ans: Windows**

290. Dynamic memory is allocated from the

- ❖ Cache

# AL-JUNAID TECH INSTITUTE

❖ Paging file

❖ Stack

❖ Static memory

291. Which of the following is recommended to use while dealing with memory mapped file to look for EXCEPTION\_IN\_PAGE\_ERROR exception?

❖ ESH exception handling

❖ SHE exception handling

❖ HE exception handling

❖ HES exception handling

292. To create a file mapping object, we have to declare \_\_\_\_\_ maximum parameters>

❖ 4

❖ 2

❖ 6

❖ 8

293. It is not possible for a system to map a file greater than \_\_\_\_\_ Into virtual memory space, while using Win32 OS.

❖ 2GB

❖ 3MB

❖ 3GB

❖ 2MB

294. It is much \_\_\_\_\_ - to sort large data available in memory rather than in files.

❖ Harder

❖ Costly

❖ Unyielding

❖ Easier

295. qsort() is a \_\_\_\_\_ function.

❖ Standard library

❖ EXE

❖ Windows DLL

❖ User defined

296. When we create a file mapped object for sorting 1000 numbers in a file recorder will be saved in a/an \_\_\_\_\_.

❖ Heap

❖ Stack

❖ Queue

❖ Array

# AL-JUNAID TECH INSTITUTE

297. Which of the following are the number of parameters taken by MapViewFile()?

- ❖ 2
- ❖ 4
- ❖ 3
- ❖ 5

298. While using MapViewOfFile(), which of the following are the three commonly used flags?

- ❖ FILE\_WRITE, FILE\_READ, AND FILE\_ALL\_ACCESS
- ❖ FILE\_MAP\_WRITE, FILE\_MAP\_READ, AND FILE\_MAP\_ALL\_ACCESS
- ❖ MAP\_WRITE, MAP\_READ, AND MAP\_AL\_ACCESS
- ❖ WRITE, READ, AND ALL\_ACCESS

299. \_\_\_\_\_ and \_\_\_\_\_ specify the starting address of the file from where the mapping starts.

- ❖ High, low
- ❖ dwFileHigh, dwFileLow
- ❖ dwFileOffsetHigh, dwFileOffsetLow
- ❖ dbFileOffsetHigh, dbFileOffsetLow

300. unmapViewOfFile() takes \_\_\_\_\_ argument(s)

- ❖ 2
- ❖ 0
- ❖ 4
- ❖ 3

301. \_\_\_\_\_ Is the API for file mapping objects.

- ❖ MakeFileMapping()
- ❖ CreateFileMapping()
- ❖ FilecreateMapping()
- ❖ Create\_file Mapping()

302. While using CreateFileMapping(), \_\_\_\_\_ refers to the paging file.

- ❖ LPSECURITY\_ATTRIBUTES
- ❖ PSECURITY\_ATTRIBUTES
- ❖ INVALID\_HANDLE\_VALUES
- ❖ INVALID\_VALUES

303. While using CreatFileMapping(), \_\_\_\_\_ allows the mapping object to be secured.

- ❖ LPSECURITY\_ATTRIBUTES
- ❖ PSECURITY\_ATTRIBUTES
- ❖ INVALID\_HANDLE\_VALUES

# AL-JUNAID TECH INSTITUTE

❖ INVALID\_VALUES

304. While using CreateFileMapping(), setting IpMapName to \_\_\_\_\_ disables the map sharing.

❖ 0

❖ 1

❖ -1

❖ **NULL**

305. DLL stand for

❖ Direct layout library

❖ **Dynamic link library**

❖ Dynamic layout library

❖ Direct link library

306. The approach to gather all the source code and library functions after encapsulation into a single executable file, is called as \_\_\_\_\_

❖ Process linking

❖ **Static linking**

❖ Dynamic linking

❖ Thread linking

307. Each DLL program will have its own copy of \_\_\_\_\_ variables.

❖ **Global**

❖ Local

❖ Dynamic

❖ Static

308. In \_\_\_\_\_ operating system DLLs are used to invoke all kernel services.

❖ **Windows**

❖ Unix

❖ Linux

❖ Solaris

309. In DLLs the executable library files are linked at \_\_\_\_\_ time.

❖ Link

❖ Run

❖ **Compile**

❖ Load

310. The entry point in DLL defined structure (DWORD) \_\_\_\_\_ values.

❖ 8

❖ **4**

❖ 2

# AL-JUNAID TECH INSTITUTE

❖ 16

311. ReadFile() and writeFile() functions perform much \_\_\_\_\_ than memory mapped file processing

❖ Slower

❖ Faster

❖ Convenient

❖ Nimble

312. Which of the following controls the paging file?

❖ The pager

❖ Direct memory access

❖ Memory mapped I/o

❖ Virtual memory management system

313. While using memory mapped I/O there is/are \_\_\_\_\_ to manage buffers for repetitive operation on the file operations.

❖ Needed

❖ Not needed

❖ Useful

❖ Mandatory

314. In order to make a program more efficient, \_\_\_\_\_ heap(s) may be required.

❖ partial

❖ only one

❖ several

❖ Minimum number of

315. There are \_\_\_\_\_ parameters taken by the HeapCreate() API.

❖ 3

❖ 2

❖ 1

❖ 4

316. The parameter "flOptions" in the HeapCreate() API is a combination of \_\_\_\_\_ flags.

❖ 2

❖ 4

❖ 3

❖ 1

317. A process can have \_\_\_\_\_ heap(s).

❖ Only two

❖ At the most one

# AL-JUNAID TECH INSTITUTE

❖ only one

❖ Many

318. \_\_\_\_\_ API is used to create a new heap.

❖ createHeap()

❖ HeapCreate()

❖ BuildHeap()

❖ NewHeap()

319. If threads have separate memory space, then it will reduce \_\_\_\_\_

❖ Memory contention

❖ Access speed

❖ Direct memory access

❖ Memory density

320. \_\_\_\_\_ is an appropriate API to dispose-off a heap handle.

❖ shudderHandle()

❖ DestroyHandle()

❖ DeleteHeap()

❖ HeapDestroy()

