

Тесты по курсу «Операционные системы»  
для студентов 2 курса специальности прикладная математика  
Побегайло А. П.

**Блок 1 – выбрать 1 вопрос**

1. Физические устройства, из которых состоит компьютер, называются:
  - потребляемыми ресурсами;
  - логическими ресурсами;
  - монопольными ресурсами;
  - разделяемыми ресурсами.
  - + **аппаратными или физическими ресурсами;**
2. Программы и данные, которые хранятся в памяти компьютера, называются:
  - потребляемыми ресурсами;
  - физическими ресурсами;
  - монопольными ресурсами;
  - + **информационными или логическими ресурсами;**
  - разделяемыми ресурсами.
3. Операционная система - это комплекс программ, которые обеспечивают:
  - надежность работы компьютера;
  - графический интерфейс для пользователя;
  - + **доступ пользователей к ресурсам компьютера;**
  - хранение данных и программ;
  - безопасность работы с компьютером.
4. Операционная система – это комплекс программ, которые обеспечивают:
  - оптимизацию ресурсов компьютера;
  - сохранение ресурсов компьютера;
  - + **разделение ресурсов компьютера между пользователями;**
  - тестирование ресурсов компьютера;
  - трассировку ресурсов компьютера.
5. Программа, которая работает под управлением операционной системы, называется:
  - + **пользовательской программой;**
  - потоком;
  - исполняемым файлом;
  - загрузочным файлом;
  - объектным файлом.
6. Если под управлением операционной системы, может одновременно исполняться только одна программа, то такая операционная система называется:
  - однопроцессорной;
  - реального времени;
  - надежной;
  - однооконной;
  - + **однопрограммной.**
7. Если под управлением операционной системы, может одновременно исполняться несколько пользовательских программ, то такая операционная система называется:
  - надежной;

- + **мультипрограммной;**
- мультипроцессорной;
- реального времени;
- мультиоконной.

8. Если операционная система может работать только на компьютере с одним процессором, то такая операционная система называется:

- реального времени;
- + **однопроцессорной;**
- одnodисковой;
- однооконной;
- однопрограммной.

9. Если операционная система может работать на компьютере с несколькими процессорами, то такая операционная система называется:

- мультидисковой;
- мультипрограммной;
- + **мультипроцессорной;**
- реального времени;
- мультиоконной.

10. Операционная система, предназначенная для управления объектами в режиме реального времени, называется:

- надежной;
- объектно-ориентированной;
- промышленной;
- + **реального времени;**
- управляющей.

11. Доступ к объектам Windows выполняется посредством:

- ссылок;
- указателей;
- + **дескрипторов, которые имеют тип HANDLE;**
- обработчика исключений;
- базового сервиса операционной системы.

12. После завершения работы с объектом его дескриптор нужно закрыть, используя функцию:

- Close;
- Destroy;
- CloseObject;
- + **CloseHandle;**
- Destruct.

## **Блок 2 – выбрать 1 вопрос**

13. Потокom управления называется:

- + **последовательность инструкций, исполняемых во время работы программы;**
- поток команд, которые исполняет процессор;
- последовательность управляющих инструкций, которые исполняет процессор;
- поток управляющих сигналов от внешнего устройства;
- поток сигналов прерывания от контроллера прерываний.

14. Контекстом потока называется:

- команды, исполняемые потоком;
- объекты синхронизации, доступные в потоке;
- переменные, к которым поток не имеет доступа;
- данные, которые поток изменяет;
- + **содержимое памяти, к которой поток имеет доступ.**

15. Поток находится в состоянии готовности, если:

- проинициализированы все переменные, доступные потоку;
- + **потоку доступны требуемые ресурсы, но не выделен квант процессорного времени;**
- потоку доступны требуемые ресурсы и выделен квант процессорного времени;
- потоку не доступны требуемые ресурсы и выделен квант процессорного времени;
- поток ждет сигнал на начало работы.

16. Поток находится в состоянии «блокирован», если:

- не проинициализированы переменные, доступные потоку;
- потоку доступны требуемые ресурсы и не выделен квант процессорного времени;
- + **потоку не доступны требуемые ресурсы и не выделен квант процессорного времени;**
- потоку не доступны требуемые ресурсы и выделен квант процессорного времени;
- поток ждет сигнал на начало работы.

17. Поток находится в состоянии выполнения, если:

- проинициализированы все переменные, доступные потоку;
- + **потоку доступны требуемые ресурсы и выделен квант процессорного времени;**
- потоку доступны требуемые ресурсы и не выделен квант процессорного времени;
- потоку не доступны требуемые ресурсы и выделен квант процессорного времени;
- поток получил сигнал на начало работы.

18. Исполняемое на компьютере приложение вместе со всеми ресурсами, требуемыми для его исполнения, называется:

- потоком;
- пользователем;
- оператором;
- + **процессом;**
- процедурой.

19. Все ресурсы, доступные процессу, называются:

- + **контекстом процесса;**
- монопольными ресурсами;
- системными ресурсами;
- потребляемыми ресурсами;
- повторно-используемыми ресурсами.

20. Какой из следующих ресурсов должен обязательно принадлежать процессу:

- принтер;
- дисплей;
- + **главный поток;**
- мьютекс;

- клавиатура.

21. Какой из следующих ресурсов должен обязательно принадлежать процессу:

+ **адресное пространство;**

- дисплей;
- мьютекс;
- мышка;
- клавиатура.

### **Блок 3 – выбрать 1 вопрос**

22. В операционных системах Windows функция CreateThread создает:

- процесс;
- консоль;
- файл;
- + **поток;**
- программу.

23. В операционных системах Windows функция ExitThread завершает:

- процесс;
- консольное приложение;
- + **поток;**
- программу;
- графическое приложение.

24. В операционных системах Windows функция TerminateThread аварийно завершает:

- процесс;
- консольное приложение;
- программу;
- графическое приложение;
- + **поток.**

25. В операционных системах Windows функция SuspendThread:

- завершает исполнение потока;
- + **приостанавливает исполнение потока;**
- аварийно завершает исполнение потока;
- возобновляет исполнение потока;
- задерживает исполнение потока на заданный интервал времени.

26. В операционных системах Windows функция ResumeThread:

- завершает исполнение потока;
- приостанавливает исполнение потока;
- аварийно завершает исполнение потока;
- + **возобновляет исполнение потока;**
- запускает поток.

27. В операционных системах Windows функция Sleep:

- завершает исполнение потока;
- приостанавливает исполнение потока;
- аварийно завершает исполнение потока;
- возобновляет исполнение потока;
- + **задерживает исполнение потока на заданный интервал времени.**

28. В операционных системах Windows функция CreateProcess создает:

- поток;
- + **процесс;**
- консоль;
- файл;
- программу.

29. В операционных системах Windows функция ExitProcess завершает:

- + **процесс;**
- консольное приложение;
- поток;
- программу;
- графическое приложение.

30. В операционных системах Windows функция TerminateProcess аварийно завершает:

- консольное приложение;
- программу;
- + **процесс;**
- графическое приложение;
- поток.

31. Объект ядра операционной системы Windows называется наследуемым, если:

- он наследует свойства объектов предков;
- его свойства наследуют объекты потомки;
- + **к нему имеют доступ дочерние процессы;**
- к нему имеет доступ любой процесс;
- к нему имеют доступ все потоки процесса.

32. Изменить свойства наследования объекта Windows можно, используя функцию:

- ChangeHandleInformation;
- ModifyHandleInformation;
- + **SetHandleInformation;**
- ResetHandleInformation.
- GetHandleInformation.

33. Определить, является ли объект Windows наследуемым можно, используя функцию:

- ViewHandleInformation;
- ShowHandleInformation;
- KnowHandleInformation;
- SetHandleInformation;
- + **GetHandleInformation.**

#### **Блок 4 – выбрать 1 вопрос**

34. Процесс, который выполняется в случае, если нет других активных пользовательских процессов, называется:

- нормальным;
- реального времени;
- рабочим;
- + **фоновым;**
- драйвером.

35. В операционных системах Windows для изменения приоритета процесса используется функция:

- ChangeProcessPriority;
- + **SetPriorityClass;**
- SetProcessPriority;
- ChangePriorityClass;
- ChangePriority.

36. В операционных системах Windows определить приоритет процесса можно при помощи функции:

- GetProcessPriority;
- + **GetPriorityClass;**
- GetProcessPriority;
- ViewPriorityClass;
- ViewPriority.

37. Приоритет потока, который учитывается операционной системой Windows при выделении потокам квантов процессорного времени, называется:

- низшим;
- реальным;
- высшим;
- + **базовым;**
- нормальным.

38. В операционных системах Windows для изменения уровня приоритета потока используется функция:

- ChangeThreadPriority;
- + **SetThreadPriority;**
- SetBasePriority;
- ChangeBasePriority;
- ModifyThreadPriority.

39. В операционных системах Windows определить уровень приоритета потока можно при помощи функции:

- + **GetThreadPriority;**
- GetBasePriority;
- SetThreadPriority;
- ViewThreadPriority;
- ShowThreadPriority.

40. Процессы обслуживаются в порядке их поступления на обработку (постановки в очередь). Такая стратегия планирования непрерываемых процессов называется:

- SPN (shortest process next);
- + **FCFS (first come – first served);**
- RR (round robin);
- SRT (shortest remaining time);
- SMP (symmetrical multiprocessing).

41. Для исполнения из очереди выбирается процесс с наименьшим ожидаемым временем исполнения. Такая стратегия планирования непрерываемых процессов называется:

- + **SPN (shortest process next);**

- FCFS (first come – first served);
- RR (round robin);
- SRT (shortest remaining time);
- SMP (symmetrical multiprocessing).

42. Прерванные процессы ставятся в конец очереди. Очередь обслуживается по алгоритму FIFO. Такая стратегия планирования прерываемых процессов называется:

- SPN (shortest process next);
- FCFS (first come – first served);
- + **RR (round robin);**
- SRT (shortest remaining time);
- SMP (symmetrical multiprocessing).

43. Для исполнения из очереди выбирается процесс с наименьшим ожидаемым временем завершения работы. Такая стратегия планирования прерываемых процессов называется:

- SPN (shortest process next);
- FCFS (first come – first served);
- RR (round robin);
- + **SRT (shortest remaining time);**
- SMP (symmetrical multiprocessing);

#### Блок 5 – выбрать 1 вопрос

44. Действие, которое не прерывается во время своего исполнения и контекст которого изменяется только самим действием, называется:

- командой;
- инструкцией;
- критической секцией;
- + **атомарным;**
- мьютексом.

45. Синхронизацией параллельных процессов называется:

- обмен данными между параллельными процессами;
- обмен управляющими сигналами между параллельными процессами;
- + **упорядочивание управляющих сигналов, которыми обмениваются параллельные процессы;**
- ожидание управляющего сигнала;
- получение управляющего сигнала.

46. Атомарное действие, которое выполняется только при условии наступления некоторого события, называется:

- + **условным;**
- необходимым;
- достаточным;
- простым;
- сложным.

47. Безусловное выполнение атомарного действия называется:

- критической секцией;
- + **взаимным исключением;**
- мьютексом;
- событием;

- синхронизацией.

48. Код, исполняемый внутри безусловного атомарного действия, называется:

- + **критической секцией;**
- взаимным исключением;
- мьютексом;
- событием;
- синхронизацией.

49. Условное атомарное действие, которое содержит только условие, называется:

- событием;
- мьютексом;
- взаимным исключением;
- + **условной синхронизацией;**
- булевым выражением.

50. Требование безопасности к решению задачи взаимного исключения формулируется следующим образом:

- поток может находиться внутри критической секции ограниченное время;
- + **в любой момент времени в критической секции может находиться только один поток;**
- поток ждет входа в критическую секцию конечное время;
- любой поток получает доступ в критическую секцию за ограниченное время;
- только один поток может использовать разделяемый ресурс.

51. Требование поступательности к решению задачи взаимного исключения формулируется следующим образом:

- любой поток получает доступ в критическую секцию за ограниченное время;
- в любой момент времени в критической секции может находиться только один поток;
- + **любой поток может находиться в критической секции ограниченное время;**
- любой поток может войти в критическую секцию;
- только один поток может использовать разделяемый ресурс.

52. Требование справедливости к решению задачи взаимного исключения формулируется следующим образом:

- любой поток может находиться в критической секции ограниченное время;
- в любой момент времени в критической секции может находиться только один поток;
- любой поток может войти в критическую секцию;
- + **любой поток получает доступ в критическую секцию за ограниченное время;**
- только один поток может использовать разделяемый ресурс.

53. Цикл ожидания while с атомарной командой микропроцессора, который ждёт разрешения на вход в критическую секцию, называется:

- событием;
- мьютексом;
- семафором;
- тупиком;
- + **активным ожиданием.**



54. Цикл ожидания while с атомарной командой микропроцессора, который ждёт разрешения на вход в критическую секцию, называется:

- + **спин-локом;**
- мьютексом;
- замком;
- тупиком;
- событием.

55. Если очередь семафора обслуживается по алгоритму FIFO, то семафор называется:

- считающим;
- бинарным;
- слабым;
- + **сильным;**
- примитивным.

56. Семафор, который может принимать только значения 0 или 1 называется:

- считающим;
- + **бинарным;**
- слабым;
- сильным;
- примитивным.

57. Семафор, который может принимать положительные целочисленные значения, называется:

- + **считающим;**
- бинарным;
- слабым;
- сильным;
- примитивным.

## **Блок 6 – выбрать 1 вопрос**

58. Поток находится в тупике, если он:

- + **ждёт событие, которое никогда не произойдет;**
- ждёт освобождение ресурса, занятого другим процессом;
- ждёт освобождения мьютекса;
- ждёт, пока семафор примет положительное значение;
- ждёт входа в критическую секцию.

59. Ресурс, который может использоваться одновременно несколькими потоками, называется:

- перераспределяемым;
- повторно-используемым;
- системным;
- + **совместно-используемым;**
- монопольным.

60. Ресурс, который может использоваться одновременно только одним потоком, называется:

- перераспределяемым;
- повторно-используемым;
- + **монопольным;**

- системным;
- совместно-используемым.

61. Ресурс, который может быть отобран у потока и перераспределен другому потоку, называется:

- повторно-используемым;
- монопольным;
- системным;
- + **перераспределяемым;**
- потребляемым.

62. Ресурс, который перестает существовать после его использования потоком, называется:

- повторно-используемым;
- монопольным;
- системным;
- перераспределяемым;
- + **потребляемым.**

63. Процесс, в котором потоки используют только повторно используемые ресурсы, находится в тупике тогда и только тогда, когда:

- граф распределения ресурсов этого процесса содержит узел;
- граф распределения ресурсов этого процесса не содержит узлов;
- + **граф распределения ресурсов этого процесса содержит цикл;**
- граф распределения ресурсов этого процесса не содержит циклов;
- граф распределения ресурсов этого процесса является сильно связным.

64. Процесс, в котором потоки используют только потребляемые ресурсы, находится в тупике тогда и только тогда, когда:

- + **граф распределения ресурсов этого процесса содержит узел;**
- граф распределения ресурсов этого процесса не содержит узлов;
- граф распределения ресурсов этого процесса содержит цикл;
- граф распределения ресурсов этого процесса не содержит циклов;
- граф распределения ресурсов этого процесса является сильно связным.

65. Контрольной точкой потока называется точка, в которой:

- останавливается отладчик;
- значение переменной изменяется отладчиком;
- происходит прерывание программы;
- выбрасывается исключение;
- + **запоминается состояние контекста потока.**

66. Транзакцией называется:

- завершение выполнения функции;
- завершение обработки исключения;
- переход потока из одного состояния в другое;
- отработка потоком кванта процессорного времени;
- + **изменение контекста потока между двумя контрольными точками.**

67. Откатом называется:

- повторный запуск программы;
- повторный запуск транзакции;
- замена транзакции;

- + **отмена транзакции;**
- запрещение транзакции.

## **Блок 7 – выбрать 1 вопрос**

68. В операционных системах Windows объект CRITICAL\_SECTION служит для решения:
- + **задачи взаимного исключения для потоков, работающих в контексте одного процесса;**
  - задачи взаимного исключения для потоков, работающих в контексте разных процессов;
  - задачи взаимного исключения для потоков, работающих в контексте как одного, так и разных процессов;
  - задачи условной синхронизации для потоков, работающих в контексте одного процесса;
  - задачи условной синхронизации для потоков, работающих в контексте разных процессов.
69. Объекты ядра Windows, которые могут находиться в одном из двух состояний (сигнальном или несигнальном), называются объектами:
- взаимного исключения;
  - сигнальными;
  - системными;
  - условной синхронизации;
  - + **синхронизации.**
70. Поток переходит в сигнальное состояние, когда он:
- начинается;
  - + **завершается;**
  - прерывается;
  - блокируется;
  - засыпает.
71. Процесс переходит в сигнальное состояние, когда он:
- прерывается;
  - блокируется;
  - начинается;
  - + **завершается;**
  - засыпает.
72. Функции ожидания ждут перехода объекта или объектов синхронизации в состояние:
- + **сигнальное;**
  - несигнальное;
  - синхронное;
  - асинхронное;
  - готовности.
73. В операционных системах Windows для ожидания перехода объекта синхронизации в сигнальное состояние используется функция:
- Wait;
  - Signal;
  - + **WaitForSingleObject;**
  - Resume;

- Synchronize.

74. В операционных системах Windows для инициализации критической секции служит функция:

- DeleteCriticalSection;
- EnterCriticalSection;
- TryEnterCriticalSection;
- + **InitializeCriticalSection;**
- LeaveCriticalSection.

75. В операционных системах Windows для входа в критическую секцию служит функция:

- DeleteCriticalSection;
- + **EnterCriticalSection;**
- TryEnterCriticalSection;
- InitializeCriticalSection;
- LeaveCriticalSection.

76. В операционных системах Windows для попытки входа в критическую секцию служит функция:

- DeleteCriticalSection;
- EnterCriticalSection;
- + **TryEnterCriticalSection;**
- InitializeCriticalSection;
- LeaveCriticalSection.

77. В операционных системах Windows для выхода из критической секции служит функция:

- DeleteCriticalSection;
- EnterCriticalSection;
- TryEnterCriticalSection;
- InitializeCriticalSection;
- + **LeaveCriticalSection.**

78. В операционных системах Windows для разрушения критической секции служит функция:

- + **DeleteCriticalSection;**
- EnterCriticalSection;
- TryEnterCriticalSection;
- InitializeCriticalSection;
- LeaveCriticalSection.

79. В операционных системах Windows мьютекс переводится в несигнальное состояние функцией:

- + **WaitForSingleObject;**
- CloseMutex;
- WaitMutex;
- ReleaseMutex;
- OpenMutex.

80. В операционных системах Windows мьютекс переводится в сигнальное состояние функцией:

- WaitForSingleObject;

- CloseMutex;
- WaitMutex;
- + **ReleaseMutex;**
- OpenMutex.

81. В операционных системах Windows событие переводится в несигнальное состояние функцией:

- SetEvent;
- CloseEvent;
- WaitEvent;
- + **ResetEvent;**
- OpenEvent.

82. В операционных системах Windows событие переводится в сигнальное состояние функцией:

- + **SetEvent;**
- CloseEvent;
- WaitEvent;
- ResetEvent;
- OpenEvent.

83. В операционных системах Windows значение семафора уменьшается на единицу посредством функции:

- ReleaseSemaphore;
- DecreaseSemaphore;
- DeleteSemaphore;
- WaitSemaphore;
- + **WaitForSingleObject.**

84. В операционных системах Windows значение семафора можно увеличить посредством функции:

- + **ReleaseSemaphore;**
- IncreaseSemaphore;
- AddSemaphore;
- WaitSemaphore;
- WaitForSingleObject.

## **Блок 8 – выбрать 1 вопрос**

85. Если данные могут передаваться по каналу только в одном направлении, то такой канал называется:

- именованным;
- анонимным;
- дуплексным;
- + **полудуплексным;**
- почтовым ящиком.

86. Если данные могут передаваться по каналу в двух направлениях, то такой канал называется:

- именованным;
- анонимным;
- + **дуплексным;**

- полудуплексным;
- почтовым ящиком.

87. Если при передаче сообщений в функциях send и receive явно указываются процессы отправитель и получатель, то такая адресация процессов называется:

- + **прямой;**
- непосредственной;
- косвенной;
- относительной;
- именованной.

88. Если при передаче сообщений в функциях send и receive указываются не адреса процессов, а имя канала передачи данных, то такая адресация процессов называется:

- прямой;
- непосредственной;
- + **косвенной;**
- относительной;
- именованной.

89. Набор правил, по которым устанавливаются связи и передаются данные между процессами, называется:

- интерфейсом;
- + **протоколом;**
- прототипом;
- сигнатурой;
- артефактом.

90. Если поток отправитель, отправив сообщение, блокируется до получения этого сообщения потоком получателем, то такое отправление сообщения называется:

- прямым;
- косвенным;
- + **синхронным;**
- асинхронным;
- полудуплексным.

91. Если поток отправитель, отправив сообщение, продолжает свою работу, то такое отправление сообщения называется:

- прямым;
- косвенным;
- синхронным;
- + **асинхронным;**
- дуплексным.

92. Если поток получатель, вызвавший функцию receive для получения сообщения, блокируется до тех пор, пока не получит сообщение, то такое получение сообщения называется:

- прямым;
- косвенным;
- + **синхронным;**
- асинхронным;
- полудуплексным.

93. Рандеву называется:

- асинхронный обмен данными в случае прямой адресации процессов;
- асинхронный обмен данными в случае косвенной адресации процессов;
- + **синхронный обмен данными в случае прямой адресации процессов;**
- синхронный обмен данными в случае косвенной адресации процессов;
- обмен данными между асинхронными процессами.

94. В операционных системах Windows анонимный канал создается посредством функции:

- CreateFile;
- CreateAnonymousPipe;
- CreateNamedPipe;
- + **CreatePipe;**
- Create.

95. В операционных системах Windows именованный канал создается посредством функции:

- Create;
- CreateFile;
- CreatePipe;
- CreateAnonymousPipe;
- + **CreateNamedPipe.**

96. В операционных системах Windows соединение сервера именованного канала с клиентом этого канала выполняется посредством функции:

- Connect;
- CreateFile\$
- ConnectFile;
- ConnectPipe;
- + **ConnectNamedPipe.**

97. В операционных системах Windows определение клиентом свободного экземпляра именованного канала выполняется посредством функции:

- Connect;
- ConnectPipe;
- ConnectNamedPipe;
- WaitPipe;
- + **WaitNamedPipe.**

98. В операционных системах Windows соединение клиента именованного канала с сервером по экземпляру этого канала выполняется посредством функции:

- Connect;
- ConnectPipe;
- ConnectNamedPipe;
- + **CreateFile;**
- OpenPipe.

99. В операционных системах Windows разрыв сервером связи с клиентом по экземпляру именованного канала выполняется посредством функции:

- Disconnect;
- DisconnectPipe;
- DisconnectFile;

- + **DisconnectNamedPipe;**
- DisconnectWithPipe.

## **Блок 9 – выбрать 1 вопрос**

100. Исполнительным устройством называется устройство, которое:
- загружает исполняемые файлы в оперативную память;
  - контролирует работу других устройств;
  - исполняет пользовательские программы;
  - исполняет системные программы;
  - + **функционально дополняет микропроцессор.**
101. Контроллеры – это устройства, которые:
- контролируют исполнение программ;
  - контролируют работу центрального процессора;
  - управляют центральным процессором;
  - + **управляют другими устройствами;**
  - интерпретируют команды перехода.
102. Устройство управления центрального процессора выполняет:
- арифметические операции с плавающими числами;
  - арифметические операции с целыми числами;
  - логические операции;
  - + **команды перехода;**
  - команды над строками.
103. Контекстом процессора называется:
- состояние оперативной памяти;
  - состояние исполняемой процессором команды;
  - + **состояние регистров процессора;**
  - состояние управляющего устройства;
  - состояние арифметико-логического устройства.
104. Точкой прерывания программы называется точка, в которой происходит:
- исполнение команды перехода;
  - возврат из подпрограммы;
  - вызов подпрограммы;
  - синхронизация потоков;
  - + **перестановка контекста процессора.**
105. Процесс перестановки контекста процессора называется:
- сбросом контекста процессора;
  - сбросом регистров процессора;
  - + **прерыванием программы;**
  - прерыванием процессора;
  - процессом обработки прерывания.
106. Контроллер прерываний обрабатывает сигналы прерывания от:
- шины данных;
  - адресной шины;
  - центрального процессора;
  - + **внешних устройств;**



- оперативной памяти.

107. Внешние прерывания могут прервать исполнение программы:

- только после завершения операции ввода данных;
- только после завершения операции вывода данных;
- только при завершении исполнения команды микропроцессора;
- + в любой точке прерывания;**
- во время исполнения любой команды микропроцессора.

108. По второму сигналу подтверждения прерывания по линии INTA от центрального процессора контроллер прерываний:

- передает управление программе обработки прерывания;
- передает управление микропроцессору;
- + устанавливает на адресную шину адрес программы обработки прерывания;**
- устанавливает на шину данных код прерывания;
- завершает свою работу.

109. На вход NMI микропроцессора поступает сигнал прерывания, который:

- маскируется;
- + не маскируется;**
- блокируется контроллером прерываний;
- блокируется микропроцессором;
- блокируется внешним устройством.

## **Блок 10 – выбрать 1 вопрос**

110. Память, к которой процессор может непосредственно обращаться, используя адресную шину и шину данных, называется:

- регистром;
- виртуальной памятью;
- + физической памятью;**
- накопителем на жестких дисках;
- магнитной лентой.

111. Память, к которой имеет доступ процесс, называется:

- + логической памятью процесса;**
- физической памятью процесса;
- виртуальной памятью процесса;
- реальной памятью процесса;
- регистром процесса.

112. Блоки одинаковой длины, на которые разбивают виртуальную память, называются:

- блоками виртуальной памяти;
- + страницами виртуальной памяти;**
- сегментами виртуальной памяти;
- адресами виртуальной памяти;
- файлами подкачки.

113. Файлы, в которых хранятся страницы виртуальной памяти, называются:

- файлами хранения виртуальных страниц;
- файлами загрузки виртуальных страниц;
- файлами отгрузки виртуальных страниц;

- + **файлами подкачки виртуальных страниц;**
- файлами замены виртуальных страниц.

114. Алгоритм FIFO для выталкивания страницы из реальной памяти на диск заключается в следующем:

- на диск выталкивается виртуальная страница, которая в последнее время не использовалась;
- на диск выталкивается виртуальная страница, которая наименее часто используется;
- на диск выталкивается виртуальная страница, которая дольше всего не использовалась;
- + **на диск выталкивается виртуальная страница, которая была загружена первой;**
- на диск выталкивается виртуальная страница, которая была загружена последней.

115. Алгоритм LRU для выталкивания страницы из реальной памяти на диск заключается в следующем:

- на диск выталкивается виртуальная страница, которая меньше всего использовалась;
- на диск выталкивается виртуальная страница, которая в последнее время не использовалась;
- + **на диск выталкивается виртуальная страница, которая дольше всего не использовалась;**
- на диск выталкивается виртуальная страница, которая была загружена первой;
- на диск выталкивается виртуальная страница, которая была загружена последней.

116 Алгоритм NRU для выталкивания страницы из реальной памяти на диск заключается в следующем:

- на диск выталкивается виртуальная страница, которая меньше всего использовалась;
- + **на диск выталкивается виртуальная страница, которая не использовалась в заданный интервал времени;**
- на диск выталкивается виртуальная страница, которая дольше всего не использовалась;
- на диск выталкивается виртуальная страница, которая была загружена первой;
- на диск выталкивается виртуальная страница, которая была загружена последней.

117. Алгоритм LFU для выталкивания страницы из реальной памяти на диск заключается в следующем:

- + **на диск выталкивается виртуальная страница, которая меньше всего использовалась;**
- на диск выталкивается виртуальная страница, которая в последнее время не использовалась;
- на диск выталкивается виртуальная страница, которая дольше всего не использовалась;
- на диск выталкивается виртуальная страница, которая была загружена первой;
- на диск выталкивается виртуальная страница, которая была загружена последней.

118. Множество страниц виртуальной памяти, которое выделяется процессу для работы и не выгружается из оперативной памяти, называется:

- фиксированным множеством страниц процесса;
- + **рабочим множеством страниц процесса;**

- загрузочным множеством страниц процесса;
- независимым множеством страниц процесса;
- закрытым множеством страниц процесса.

119. В операционных системах Windows резервирование или распределение области виртуальной памяти выполняется посредством функции:

- + **VirtualAlloc;**
- VirtualLock;
- VirtualUnlock;
- VirtualFree;
- VirtualProtect.

120. В операционных системах Windows для блокирования страниц виртуальной памяти в реальной памяти используется функция:

- VirtualAlloc;
- + **VirtualLock;**
- VirtualUnlock;
- VirtualFree;
- VirtualProtect.

121. В операционных системах Windows для разблокирования страниц виртуальной памяти в реальной памяти используется функция:

- VirtualAlloc;
- VirtualLock;
- + **VirtualUnlock;**
- VirtualFree;
- VirtualProtect.

122. В операционных системах Windows для освобождения области виртуальной памяти используется функция:

- VirtualAlloc;
- VirtualLock;
- VirtualUnlock;
- + **VirtualFree;**
- VirtualProtect.

123. В операционных системах Windows для изменения атрибутов доступа к области виртуальной памяти используется функция:

- VirtualAlloc;
- VirtualLock;
- VirtualUnlock;
- VirtualFree;
- + **VirtualProtect.**

124. В операционных системах Windows узнать количество виртуальных страниц, которые входят в рабочее множество страниц процесса, можно посредством функции:

- ViewProcessWorkingSetSize;
- ShowProcessWorkingSetSize;
- + **GetProcessWorkingSetSize;**
- SetProcessWorkingSetSize;
- ChangeProcessWorkingSetSize.

125. В операционных системах Windows минимальный и максимальный размеры рабочего множества страниц процесса можно изменить посредством функции:

- ViewProcessWorkingSetSize;
- ShowProcessWorkingSetSize;
- GetProcessWorkingSetSize;
- + **SetProcessWorkingSetSize;**
- ChangeProcessWorkingSetSize.

126. В операционных системах Windows определить состояние области виртуальной памяти процесса можно посредством функции:

- VirtualAlloc;
- VirtualLock;
- VirtualUnlock;
- + **VirtualQuery;**
- VirtualProtect.

**На весь тест – 30 минут**

**Оценка:**

**100% - 10 баллов**

**90% - 9 баллов**

**80% - 8 баллов и т.д.**