

Жёлтый - важно.

Красное - правильные ответы, которые были в тесте (но лучше знать всё, мог забыть что-то). (+) – обознач.

Сигналы поступают на спец. **контроллёр** прерываний.

Всего существует **16 прерываний** (0-15).

Прерывания: (**на всякий случай, мб и не было в тесте этого**).

0 - Наиболее приоритетное связано с системным таймером и не как не может быть использовано другим устройством.

1 - Закреплено за контроллером клавиатуры.

2 - Имеет техническое значение, благодаря которому изначальное число прерываний было увеличено с 8 до 16.

3 - Используется вторым последовательным портом, однако если этот порт не нужен, то его можно отключить.

4 - Используется первым последовательным портом.

5 - Изначально является свободным и может использоваться различными устройствами по усмотрению пользователя или ОС.

6 - Закреплено за контроллером FDD. Не может быть использовано другими устройствами.

7 - Используется параллельным портом компьютера. Это прерывание при отключенном порте можно закрепить за другим устройством.

8 - Закреплено за часами реального времени. Это прерывание никогда не используется другими устройствами.

9 - Изначально является свободным, может использоваться платами расширения по усмотрению пользователя или ОС. Контроллер USB порта.

10 - Свободное, платы расширения.

11 - Изначально является свободным, но обычно используется видеоадаптерами. 12 - Жестко закреплено за мышью PS/2

13 - Жестко закреплено за встроенным или внешним сопроцессором.

математическим

14 - Жестко закреплено за первым каналом контроллера. Не доступно для других 10 устройств.

15 - Жестко закреплено за вторым каналом IDE.

С каждым устройством ввода вывода связана область памяти, называемая **векторами прерываний**.

Для того чтобы упорядочить обработку прерываний для каждого из них устанавливается **приоритет**.

Для правильного функционирования системы необходимо **взаимодействие** между диспетчером, выполняющим процессом и подсистемой ввода/вывода.

Обычно истечение кванта времени или завершение ввода/вывода выявляется в результате **прерывания**.

Прерывание - **временное** прекращение процесса, вызванное событием, внешним по отношению к этому процессу.

Условные переходы - изменение последовательности выполнения операторов программы в зависимости от результата проверки некоторого условия.

Безусловные переходы - переход в заданную точку программы без проверки выполнения каких-либо условий.

Прерывания распадаются на **2** основных класса:

Внешние прерывания - вызываются асинхронными событиями, которые происходят вне прерываемого процесса.

Внутренние прерывания - вызываются событиями, которые связаны с работой процессора и являются **синхронными** с его операциями.

В некоторых машинах часть команд может использовать только **ОС** (а не пользователь).

Последовательность действий **обработки прерываний**: (**будет в тесте**). (**установить последовательность**).

1. В некоторую фиксированную ячейку заносятся характеристики произошедшего прерывания.

2. Запоминается состояние прерванного процесса.

3. В «счетчик команд» заносится фиксированный адрес, который, как правило, является уникальным для каждого типа прерывания.
4. Обрабатывается прерывание.
5. Возобновляется нормальная работа.

Супервизор - программа, управляющая процессом, памятью и работой оборудования ОС.

Множественные прерывания. Прерывания происходят не всегда изолированно; они могут происходить одновременно или, по крайней мере, одновременно фиксироваться аппаратной частью ЭВМ. Прерывания могут также возникать во время обработки других прерываний.

Прерывания, класс которых имеет более низкий приоритет, могут быть **разрешены**, либо **замаскированы**. Если прерывание **разрешено**, то оно может **прервать процессор**, что обычно и происходит сразу же после того, как аппаратура зафиксирует прерывание.

Замаскированное прерывание не может прервать процессор. Оно может быть утерянным, либо ждать обработки, что означает, что оно все еще известно аппаратуре. Если замаскированное прерывание ждет обработки, то последующие размаскированные или разрешение прерываний, инициирует его обработку.

Если новое прерывание происходит во время старого, то оно будет:

1. Разрешено (+)
2. Замаскировано (+)

Если новое прерывание **разрешено**, то оно начинает обрабатываться, а программа обработки старого является в этом случае прерванным процессом. Если новое прерывание **замаскировано**, то оно может быть либо утерянным, либо ожидать обработки. Если прерывание **ожидает обработки**, то находится в этом состоянии до тех пор, пока управляющая программа не будет готова к его обработке.

Обработка **множественных прерываний** при наличии разных приоритетов облегчается присутствием аппаратных средств, позволяющих выделить разным приоритетам различных фиксированных ячеек для характеристик прерывания и для запоминания состояния процесса.

Прерывания, **ожидающие обработки** обычно хранятся в **очередях**.

Определения процесса:

1. Выполняющаяся прикладная программа пользователя. (+)
2. Экземпляр программы, выполняющийся на компьютере.
3. Объект, который можно идентифицировать и выполнять на процессоре. (+)
4. Единица активности, которую можно охарактеризовать единой цепочкой последовательных действий.

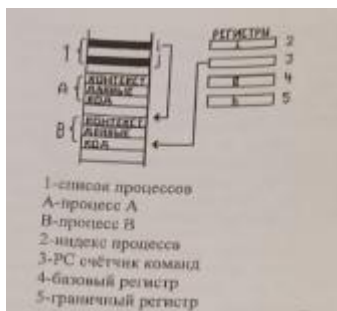
Последовательный процесс (задача) - выполнение отдельной программы и её данные на последовательном процессоре.

В качестве примера можно назвать **следующие процессы:**

1. Выполняющаяся прикладная программа пользователя.
2. Утилит.
3. Трансляция программ.
4. Компоновка, выполнение.

Мысленно процесс можно **разделить на 2 компонента:**

1. Выполняющаяся программа.
2. Данные нужные для работы.



Реализация процесса. (будет в тесте).

В активном состоянии процесс участвует в конкуренции за использование ресурса вычислительной системы, а **в пассивном** он только известен системе, но в конкуренции не участвует.

Состояния активного состояния:

1. Выполнение- все затребованные процессом ресурсы выделены. В этом состоянии в каждый момент времени может находиться только 1 процесс.
2. Готовность к выполнению - ресурсы предоставлены, тогда процесс перейдёт в состояние выполнения.
3. Блокирование или ожидание - затребованные ресурсы не могут быть предоставлены или не завершена операция вв/выв.

Дескриптор - программный канал, обеспечивающий связь между процессами.

Процесс может перейти из состояния бездействия в состояние готовности **в следующих случаях:**

1. По команде оператора. (+)
2. При выборе из очереди планировщиком.
3. По вызову из другой задачи. (+)
4. По прерыванию от внешнего инициативного устройства.
5. При наступлении запланированного времени запуска программ. (+)

Из состояния выполнения процесс может выйти по одной **из следующих причин:**

- 1) Процесс завершается, при этом он передаёт управление ОС и сообщает о своём завершении.
- 2) Процесс переводится ОС в состояние готовности к выполнению в связи с появлением более приоритетной задачи и в связи с окончанием выделенного ему кванта времени.
- 3) Процесс блокируется (переводится в состояния ожидания) либо в силу невозможности предоставить ему ресурс, запрошенный в настоящий момент.

Деструктор - специальная информационная структура процесса (или описатель задачи, блок управления задачами).

Деструктор содержит след. инфу:

1. Идентификатор процесса (+)
2. Тип или класс процесса
3. Приоритет процесса
4. Переменную состояния (+)
5. Защищённую область памяти
6. Информация о ресурсах
7. Место памяти или адрес этого места
8. Параметры времени запуска (+)

Концепция процесса:

1. Владение ресурсами
2. Планирование и выполнение

Единица диспетчеризации - **поток**.

Единица владения ресурсами - **процесс или задача**.

Многопоточность - способность ОС поддерживать в рамках одного процесса выполнение нескольких потоков.

Однопоточность - традиционный подход, при котором каждый процесс представляет собой единый поток выполнения.

Использование потоков:

- 1) создание нового потока в уже существующем процессе занимает меньше времени, чем создание нового процесса;
- 2) поток можно завершить быстрее чем процесс;
- 3) переключение потоков в рамках одного и того же процесса происходит быстрее;
- 4) при использовании потока повышается эффективность обмена информацией между двумя выполняющимися программами. (+)

Движущая сила, меняющая состояния процесса - **событие**.