**Тема 4. Взаимодействие и планирование процессов**

**Лекция 12 «Взаимодействие и планирование процессов».**

**Взаимодействие процессов.** Совместно выполняемые процессы могут быть либо независимыми, либо взаимодействующими. При взаимодействии процессов создается совместный буфер для взаимного обмена данными. Взаимодействие заключается в передаче данных между процессами или совместном использовании ресурсов и реализуется с помощью таких механизмов, как: транспортеры, очереди, сигналы, семафоры.

**Транспортеры (каналы)** – являются средством взаимодействия родственных процессов, представляют собой область памяти, имеющую файловую организацию, для которой обеспечивается запись и считывание данных в транспортере.

**Очереди** – эти механизмы могут обеспечивать передачу или использование общих данных без перемещения данных, а с передачей элемента очереди, содержащего указатель данных и объем массива данных.

**Сигналы** – являются механизмами передачи требования от одного процесса к другому на немедленное выполнение действия. Характер выполняемых действий при возникновении сигнала: обработка системной ошибки при появлении сигнала, блокирование сигнала, передача управления подпрограмме.

**Семафоры** – являются механизмами передачи сообщений от одного потока к другому о наступлении некоторого события. Различают семафоры системные и оперативной памяти.

В операционных системах OS/2 и Microsoft Windows существует специальный механизм для взаимодействия процессов в реальном масштабе времени. Механизм **DDE** (Dynamic Data Exchange – динамический обмен данными). Он стандартизирует процесс обмена командами, сообщениями и объектами для обработки между задачами. Наиболее распространенным процессом, для которого используется DDE является печать.

Другим механизмом для обмена данными является **OLE** (Object Linking and Embedding – связывание и встраивание объектов), позволяет хранить объекты, созданные одной программой, в объектах, созданных другой программой, а также редактировать (печатать) их без нарушения целостности информации и связей.

Одним из наиболее простых и интуитивных интерфейсов межпрограммного взаимодействия является буфер обмена – **Clipboard**. Буфер обмена может содержать в себе один информационный объект – фрагмент текста, рисунок и т.д. С помощью системного вызова процесс может получить копию информации, содержащейся в буфере обмена, или сам поместить объект в буфер, при этом «старое» содержимое буфера теряется.

**Планирование процессов** – это деятельность диспетчера процессов, которая обрабатывает удаление запущенного процесса из ЦП и выбор другого процесса на основе конкретной стратегии, т.е. процесс выбора – кто будет исполняться следующим и как долго это будет исполняться. Планирование процессов является неотъемлемой частью многопрограммных операционных систем. Такие операционные системы позволяют загружать в исполняемую память более одного процесса за один раз.

**Планирование** - обеспечение поочередного доступа процессов к одному процессору, а **планировщик** - отвечающая за это часть операционной системы.

**Классы планировщиков:**

**Пакетный** – ориентирован на длительные задачи, которые требуют больших вычислительных ресурсов, где не требуется частое прерывание (подразумевают обработку больших задач большими пакетами, нет ограничения на время выполнения).

**Интерактивный** – ориентирован на снижение времени отклика, т.е. чтобы система казалась «отзывчивой». Обычные абонентские системы на ПК – это интерактивные системы, когда в ответ на действие пользователя (например, перемещение мыши) ОС что-то делает. Пользователю хочется, чтобы этот ответ происходил как можно быстрее. Главное, чтобы на поступающий в систему запрос был получен максимально быстрый ответ. Запрос – это любое взаимодействие с компьютером.

**Реального времени** – специализированные класс, ориентированный на «дедлайн» – предельный срок завершения какой-либо работы. Главное, чтобы определенное действие завершалось к определенному сроку, это понятие называется «дедлайн». Поступающий запрос должен быть обработан не более, чем в определенный промежуток времени. (пример СРВ – управление ядерным реактором, в котором превышение времени отклика приведет к аварийной ситуации).

**Ситуации, когда необходимо планирование:**

1. Когда создается процесс;
2. Когда процесс завершает работу;
3. Когда процесс блокируется на операции ввода/вывода, семафоре, и т.д.;
4. При прерывании ввода/вывода.

**Алгоритм планирования** - используемый алгоритм для планирования. Бывают:

1. **Алгоритм планирования без переключений (неприоритетный)** - не требует прерывание по аппаратному таймеру, процесс останавливается только когда блокируется или завершает работу.
2. **Алгоритм планирования с переключениями (приоритетный)** - требует прерывание по аппаратному таймеру, процесс работает только отведенный период времени, после этого он приостанавливается по таймеру, чтобы передать управление планировщику.

**Очереди планирования процессов:**

* **Очередь заданий** – в этой очереди хранятся все процессы в системе.
* **Готовая очередь** – эта очередь хранит набор всех процессов, находящихся в основной памяти, готовых и ожидающих выполнения. Новый процесс всегда помещается в эту очередь.
* **Очереди устройства** – процессы, которые заблокированы из-за недоступности устройства ввода-вывода, составляют эту очередь.

**Модель процесса** бывает с двумя состояниями:

1. **Бег** (рабочее состояние)**.** Когда создается новый процесс, он входит в систему как в рабочем состоянии.
2. **Не бегать** (неработающее состояние)**.** Процессы, которые не запущены, остаются в очереди, ожидая своей очереди на выполнение. Каждая запись в очереди является указателем на определенный процесс. Очередь реализуется с помощью связанного списка. Использование диспетчера заключается в следующем. Когда процесс прерывается, этот процесс передается в очередь ожидания. Если процесс завершен или прерван, процесс отбрасывается. В любом случае диспетчер затем выбирает процесс из очереди для выполнения.