

СИСТЕМА УПРАВЛЕНИЯ ПРОЦЕССАМИ

При построении систем управления процессами в большинстве ОС принята двухуровневая схема. Это означает, что имеется два вида планировщиков процессов:

- **планировщик высокого уровня**, выполняющий **долгосрочное планирование**;
- **планировщик низкого уровня**, выполняющий **краткосрочное планирование**.

Объектом планирования является ЦП, цель планирования – обеспечение функционирования некоторого множества процессов. Каждый процесс, подконтрольный ОС, характеризуется так называемым **дескриптором**. Дескриптор – специальная информационная структура, содержащая управляющую информацию о процессе и его связи с другими процессами.

На уровень долгосрочного планирования выносятся действия редкие в системе, но требующие больших системных затрат. На уровень краткосрочного планирования выносятся частые и более короткие по длительности действия по управлению процессами.

Уровень долгосрочного планирования. На уровне долгосрочного планирования объектом управления являются не отдельные процессы, а некоторые их объединения, рассматриваемые как единое целое, объединенное единым функциональным назначением. Такие объединения процессов называются **заданиями**. Каждое задание рассматривается как независимая от других заданий деятельность. Примером задания является выполнение программы, написанной на алгоритмическом языке. Шаги задания – компиляция, компоновка и исполнение.

Задание само может рассматриваться как процесс в обобщенном смысле («суперпроцесс»), программа которого выполняется на виртуальной машине. Задание, рассматриваемое как процесс, характеризуется совокупностью состояний.

1. Состояние *порождения* – в ответ на требование исполнить задание планировщик высокого уровня создает виртуальную машину. Требование на порождение задания, выраженное в форме программы на командном языке, являются внешними по отношению к ЦП. Создание виртуальной машины заключается в следующих действиях:

- резервируются ресурсы, указанные в задании, и обеспечивается доступ к данным на внешних носителях;
- резервируется память, в которой будет размещаться программа задания;
- создается структура данных, исполняющая роль *дескриптора задания* и содержащая информацию, необходимую для управления исполнением задания;

2. Состояние *готовности* означает, что для исполнения задания предоставлены все ресурсы виртуальной машины, кроме виртуального процессора. Деятельность виртуального процессора по исполнению каждого элементарного действия задания моделируется на ЦП с помощью планировщика низкого уровня. Планировщик высокого уровня выдает заявку на порождение определенного процесса на нижнем уровне, что соответствует переводу задания из состояния готовности в состояние активности.

3. Состояние *активности* соответствует исполнению задания на виртуальном процессоре. По мере окончания исполнения очередного элементарного действия на уровне краткосрочного планирования, планировщик этого уровня передает сообщение планировщику высокого уровня, в ответ на что планировщик высокого уровня подготавливает условия для выполнения следующего элементарного действия из состава программы задания и вырабатывает новую заявку на порождение процесса на нижнем уровне.

4. Состояние *окончания* определяет следующую совокупность действий под управлением планировщика высокого уровня:

- вычисление системных затрат на выполнение задания;
- передача результирующих выходных данных на указанные носители информации;
- освобождение всех ресурсов, использованных для построения виртуальной машины, и уничтожение дескриптора задания.

Уровень краткосрочного планирования. Процесс представляет собой совокупность отдельных состояний. Переходы между этими состояниями планируются и осуществляются под управлением планировщика нижнего уровня, **называемого диспетчером задач**.

Переход в состояние *порождения* происходит при выполнении заявки, поступившей с уровня долгосрочного планирования. Порождение заключается в следующем:

- определяется имя процесса и соответствующая программа;
- обеспечивается доступ к ресурсам, которые должны быть выделены перед исполнением программы (в том числе осуществляется передача параметров);

- резервируется память, необходимая для выполнения указанного программного модуля;
- создается дескриптор процесса.

Переход процесса в состояние *завершения* происходит при выполнении заявки об окончании исполнения программы. Завершение может быть нормальным (обусловленным алгоритмом программы), либо аварийным. Завершение включает в себя действия обратные тем, которые осуществляются при порождении процесса:

- передается или сохраняется информация о результатах работы процесса тем процессам, которые ее потребляют;
- освобождаются распределенные ресурсы (передаются процессу, породившему данный);
- осуществляется уничтожение дескриптора процесса.

Уничтожение процесса связано с рядом проблем. Например, уничтожаемый процесс может быть порождающим в отношении ряда других процессов и контролировать их развитие. Что делать с «потомками»? Чаще всего уничтожается всё дерево. Иногда процесс-родитель запрещается уничтожать раньше его процессов-потомков.

Другая проблема связана с необходимостью корректно закрыть связи с другими процессами (информационные связи, совместное использование ресурсов и т.д.). Одно из решений состоит в том, что при уничтожении процесса посылаются специальные пустые сообщения тем процессам, которые ждут от него поступления информации. В случае нахождения процесса в критической области делается искусственный выход из нее.

Переходы между другими состояниями также выполняются с помощью специальных системных действий. Некоторые из них доступны самим процессам (переходы из активного состояния в другие). В других состояниях сам процесс не может выполнить каких-либо действий.

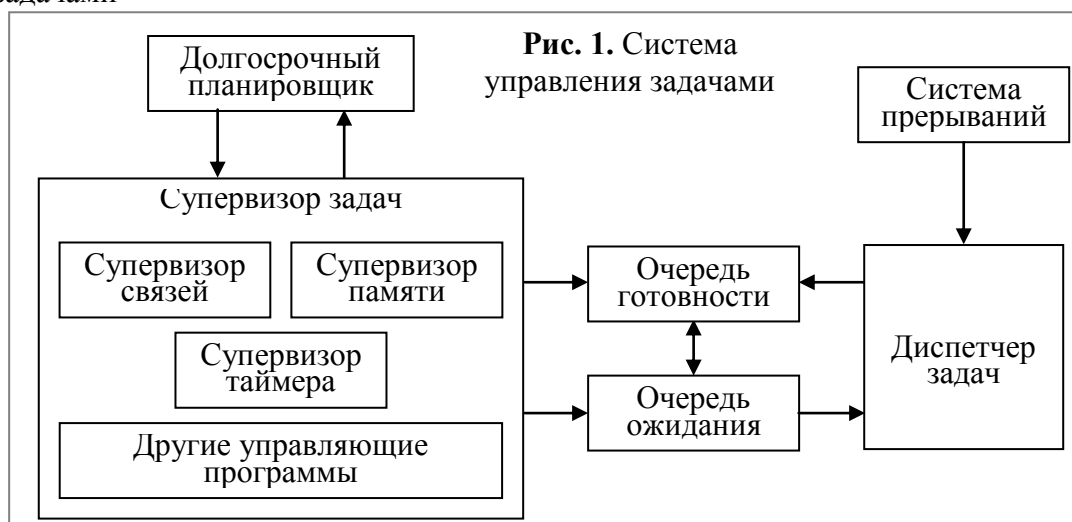
Отработка перехода из активного состояния в другие (как по инициативе самого процесса, так и по внешней причине) основана на механизме *обработки прерываний*.

Основой для перевода процесса в состояния *ожидания* и *готовности* являются соответствующие очереди. Средства по переводу процессов в состояние *ожидания* и из него в состояние *готовности* базируются на использовании средств синхронизации процессов.

Перевод процессов в активное состояние является функцией краткосрочного планировщика. Он по определенному правилу (на основе приоритетов) выбирает из очереди процессов, находящихся в состоянии готовности, один конкретный процесс и производит действия по его активизации:

- восстанавливает прерванное состояние процесса (если он прерывался);
- вносит изменения в дескриптор процесса;
- передает управление на исполнение.

На рис.1 показана реализация краткосрочного планировщика в виде системы управления задачами



Супервизор задач – это системная программа, выполняющая функции по управлению процессами. Поддерживает механизм синхронизации параллельных процессов, механизм статического и динамического

назначения приоритетов (учитываемых при распределении ЦП). В ее состав входит также *супервизор связей*, отвечающий за создание, уничтожение и перевод задач из состояния в состояние. *Супервизор памяти* выполняет распределение оперативной памяти, а *супервизор таймера* управляет обращениями к интервальному таймеру по требованиям параллельных задач.

Диспетчер задач – это системная программа, планирующая использование ЦП (совместно с программно-аппаратной системой прерываний) т.е. выполняющая функции краткосрочного планирования. Он обслуживает две очереди – *очередь готовности* и *очередь ожидания*.