**Тезаурусный словарь**

Unix — одна из первых многозадачных ОС. Каждый процесс имеет уникальный числовой идентификатор PID. Процессы в ней имеют древовидную иерархию, где корнем является процесс init c PID 1. Новый процесс можно создать системным вызовом fork, он будет являться точной копией процесса родителя. Любой процесс кроме init всегда имеет процесс родитель (атрибут PPID (англ. Parent PID)); процессы, родитель которых завершил свою работу становятся дочерними процессами init. Процессы также объединяются в группы. За управление идентификатором группы (PGID) отвечают системные вызовы setpgid и getpgid. PGID равен PID’у лидера группы. Процесс потомок наследует группу от родителя. Группы используются для управления заданиями.

Поток — определенный способ выполнения процесса. Когда один поток изменяет ресурс процесса, это изменение сразу же становится видно другим потокам этого процесса.

**Ментальная карта**

**Состояние процессов**

Операция ввода-вывода

Командные циклы

# Состояние процессов

Основной задачей процессора является выполнение машинных команд, которые находятся в основной памяти и составляют программу. В предыдущей главе объяснялось, что с целью повышения эффективности и облегчения процесса программирования процессор может одновременно выполнять несколько программ, чередуя их во времени.

Ранее упоминалось, что для каждой программы, которая должна быть выполнена, создается свой процесс, или задание. С точки зрения процесса его работа состоит в выполнении определенного набора команд; последовательность выполнения этих команд задается адресами, которые заносятся в счетчик команд. Через некоторое время счетчик команд может адресовать код других программ, которые являются частями других процессов, но с точки зрения данной программы ее выполнение состоит из последовательного выполнения ее команд.

Поведение процесса можно охарактеризовать, последовательно перечислив выполненные в ходе его работы команды. Такой перечень выполненных команд процесса называется его следом (trace). Поведение процессора можно охарактеризовать, показав, как чередуются следы различных процессов.

Рассмотрим очень простой пример. На рис.1 показано расположение в памяти трех процессов. Чтобы упростить обсуждение, предположим, что виртуальная память не используется; таким образом, все три процесса представлены программами, которые полностью загружены в основную память. Кроме этих программ в памяти находится небольшая программа-диспетчер, выполняющая переключение с одного процесса на другой. В листинге 1 показаны следы трех рассматриваемых процессов на ранних стадиях их выполнения. Представлены первые 12 выполненных команд в процессах А и С; в процессе В выполнено четыре команды, и мы считаем, что эти команды включают в себя операцию ввода-вывода, завершения которой должен ожидать процесс.

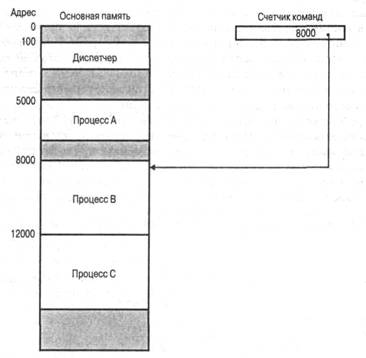
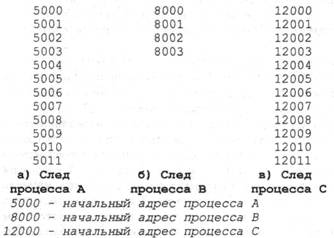


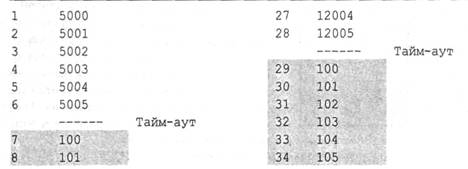
Рисунок 1 Состояние системы в момент выполнения 13-го командного цикла (см. листинг .2)

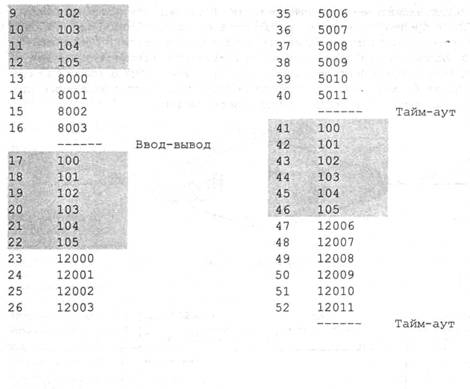
Листинг 1. Следы процессов, изображенных на рис.1



Теперь рассмотрим эти следы с точки зрения процессора. В листинге 2 показаны чередующиеся следы, получившиеся в результате выполнения первых 52 командных циклов (для удобства они пронумерованы). Предположим, что операционная система позволяет непрерывно выполнять не более шести командных циклов одного и того же процесса, после чего процесс прерывается — это предотвращает монопольное использование всего процессорного времени одним из процессов. Из листинга 2 видно, что после первых шести команд процесса А следует перерыв, в течение которого выполняется некоторый код диспетчера, состоящий из шести команд, после чего управление передается процессу В.3 Выполнив четыре команды, процесс В запрашивает операцию ввода-вывода и должен ожидать ее завершения. Поэтому процессор прекращает выполнять процесс Вис помощью диспетчера переходит к выполнению процесса С. После очередного перерыва процессор возобновляет выполнение процесса А. По истечении отведенного этому процессу времени процесс В все еще ожидает завершения операции ввода-вывода, поэтому диспетчер снова передает управление процессу С

Листинг. 2. Составной след процессов, изображенных на рис. 1





100 - начальный адрес программы-диспетчера Заштрихованные области - выполнение команд диспетчера В первом столбце указаны номера командных циклов, во втором - адреса выполняемых команд