

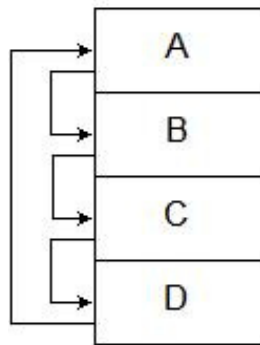
# Процессы и потоки

## Операционные системы

### Лекция №2

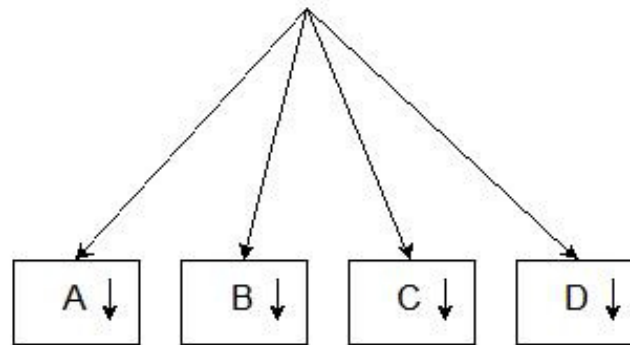
Ульяновск, УлГТУ, кафедра «Информационные системы»

Один счетчик команд



а)

Четыре счетчика команд



б)

Процесс



в)

Четыре программы, работающие в многозадачном режиме а);  
концептуальная модель четырех независимых (последовательных)  
процессов б);  
в отдельно взятый момент времени активна только одна программа в).

Основные события, приводящие к созданию процесса:

1. Инициализация системы.
2. Выполнение работающим процессом системного вызова, предназначенного для создания процесса.
3. Запрос пользователя на создание нового процесса.
4. Инициализация пакетного задания.

Основные события, приводящие к завершению процесса:

1. Обычное завершение (добровольное).
2. Завершение при возникновении ошибки (добровольное).
3. Завершение при возникновении фатальной ошибки (принудительное).
4. Завершение текущего процесса другим процессом (принудительное).



1. Процесс заблокирован в лжидании ввода
2. Диспетчер выбрал другой процесс
3. Диспетчер выбрал данный процесс
4. Входные данные стали доступны

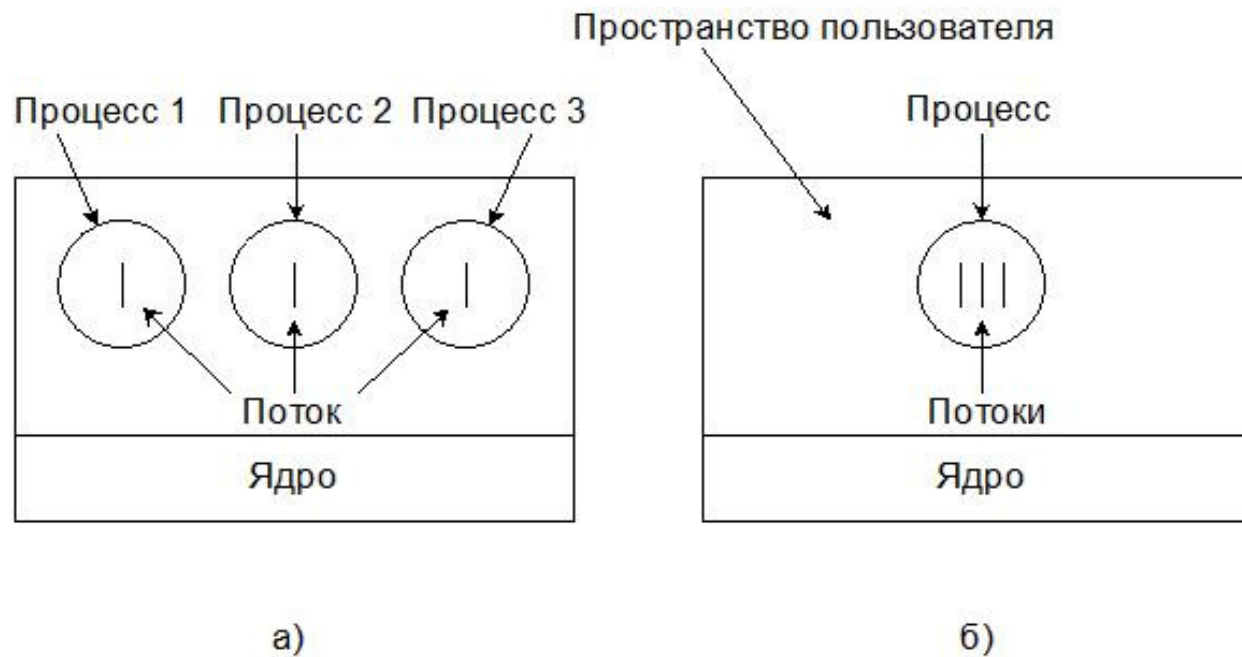
Процесс может находиться в состоянии выполнения, блокировки или готовности.

Некоторые из полей типичной записи таблицы процессов:

1. Регистры.
2. Счетчик команд.
3. Слово состояния программы.
4. Указатель стека.
5. Состояние процесса.
6. Приоритет.
7. Параметры планирования.
8. Идентификаторы процесса.
9. Родительский процесс.
10. Группа процесса.
11. Сигналы.
12. Время запуска процесса.
13. Использованное время процессора.
14. Время процессора, использованное дочерними процессами.
15. Время следующего аварийного сигнала.

Схема работы низшего уровня операционной системы при возникновении прерывания:

1. Оборудование помещает в стек счетчик команд.
2. Оборудование загружает новый счетчик команд из вектора прерывания.
3. Процедура на ассемблере сохраняет регистры.
4. Процедура на ассемблере устанавливает указатель на новый стек.
5. Запускается процедура на языке С, обслуживающая прерывание.
6. Планировщик принимает решение, какой процесс запускать следующим.
7. Процедура на языке С возвращает управление ассемблерному коду.
8. Процедура на ассемблере запускает новый процесс.



Три процесса, у каждого из которых по одному потоку а); один процесс с тремя потоками б).

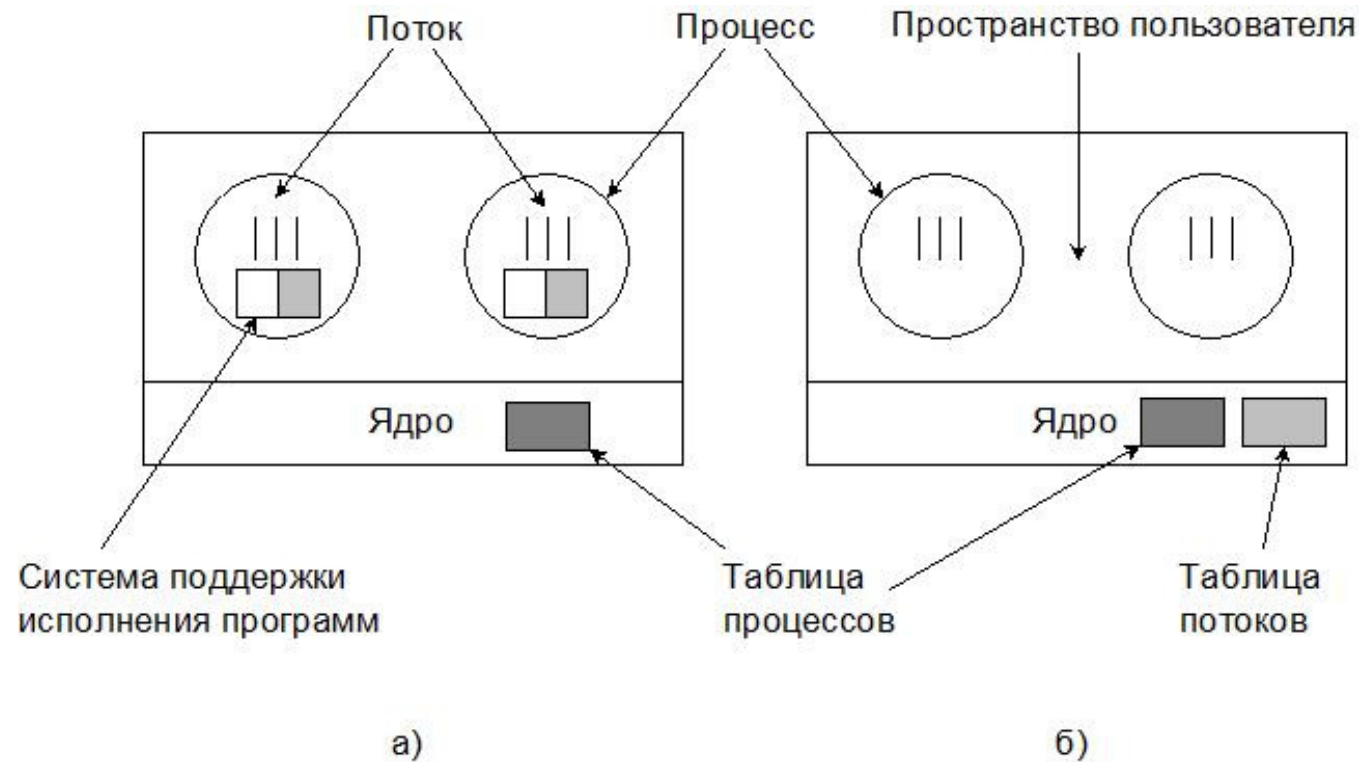
Элементы процесса:

1. Адресное пространство.
2. Глобальные переменные.
3. Открытые файлы.
4. Дочерние процессы.
5. Необработанные аварийные сигналы.
6. Сигналы и обработчики сигналов.
7. Учетная информация.

Элементы потока:

1. Счетчик команд.
2. Регистры.
3. Стек.
4. Состояние.





Набор потоков на пользовательском уровне а); набор потоков, управляемый ядром б).

**Состязательная ситуация** – ситуация, когда два или более процесса считывают или записывают какие-нибудь общие данные, а окончательный результат зависит от порядка выполнения процессов.

**Критическая область** – часть программы, использующая общие ресурсы.

Методы решения проблемы:

1. Запрещение прерываний.
2. Блокирующие переменные.
3. Строгое чередование.
4. Алгоритма Петерсона.
5. Команда TSL (Test and Set Lock).
6. Приостановка и активизация.
7. Семафоры.
8. Мьютексы.
9. Мониторы.
10. Передача сообщений.
11. Барьеры.

**Процесс, ограниченный скоростью вычислений** – основная часть процессорного времени тратится на вычисления.

**Процесс, ограниченный скоростью работы устройств ввода-вывода** – основная часть процессорного времени тратится на ожидание завершения операций ввода-вывода.

Категории алгоритмов планирования:

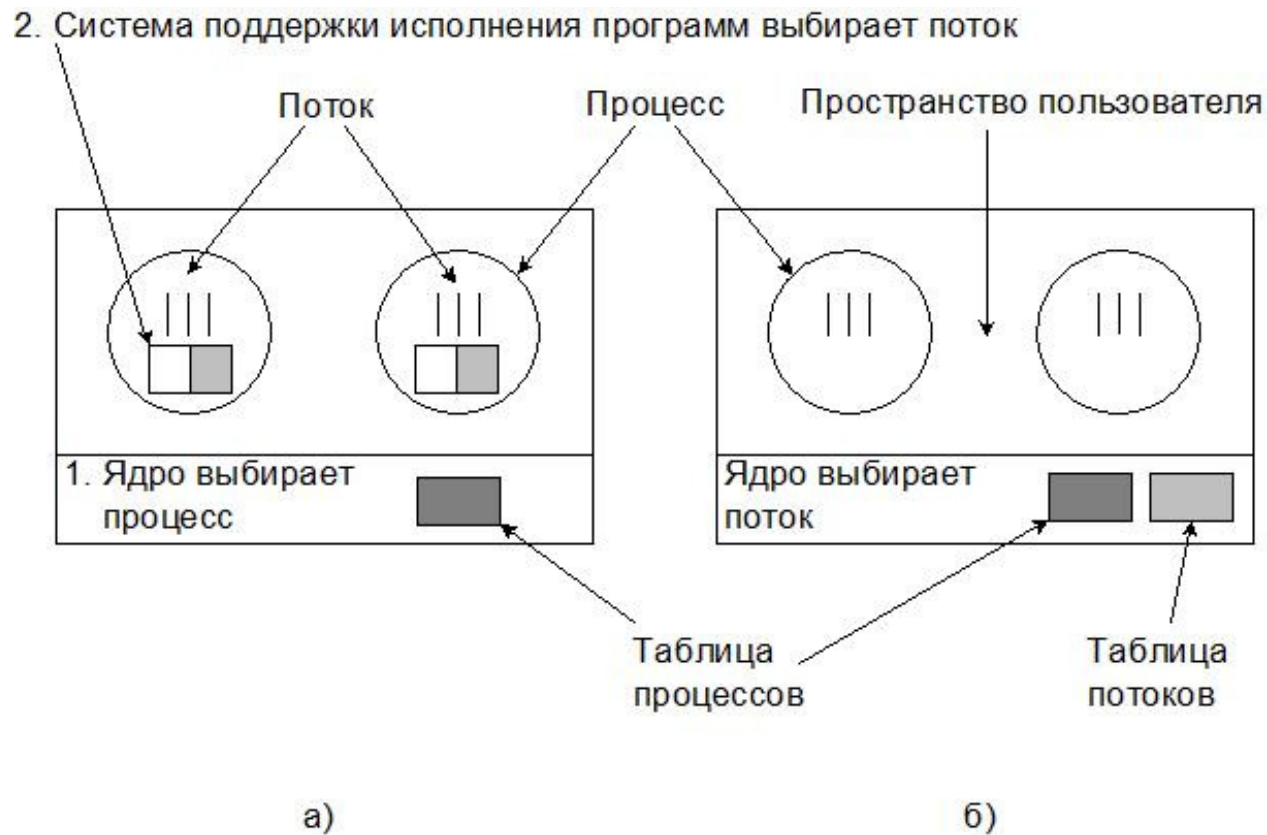
1. Пакетная:

- ▶ первым пришел – первым обслужен;
- ▶ сначала короткое задание;
- ▶ приоритет наименьшему времени выполнения.

2. Интерактивная:

- ▶ циклическое планирование;
- ▶ приоритетное планирование;
- ▶ использование нескольких очередей;
- ▶ выбор следующим самого короткого процесса;
- ▶ гарантированное планирование;
- ▶ лотерейное планирование;
- ▶ справедливое планирование.

3. Реального времени.



Планирование потоков в пользовательском пространстве а); планирование потоков в пространстве ядра.