Понятие процесса

Процесс — это программа, находящаяся в состоянии выполнения. Процесс представляет собой активную сущность, в отличие от программы как пассивного объекта (файла на диске). Процесс включает в себя:

исполняемый код программы;

текущее состояние (включая значения регистров, указатель команд и т. д.);

адресное пространство (оперативная память, используемая процессом);

набор ресурсов, выделенных процессу (например, дескрипторы файлов, устройства ввода-вывода и др.).

Операционная система управляет процессами, обеспечивая их создание, выполнение, приостановку и завершение.

Отличие процесса от программы

Программа — это статическая последовательность инструкций, находящаяся в памяти (обычно на диске).

Процесс — это динамическая активность, возникающая при выполнении программы. Один и тот же программный файл может быть основой для нескольких независимых процессов.

Состояния процесса

Процесс в течение своего жизненного цикла проходит несколько состояний. Операционная система отслеживает и управляет переходами между этими состояниями.

Основные состояния процесса:

New (Новый) — процесс только что создан, но ещё не готов к выполнению.

Ready (Готов) — процесс готов к выполнению и ожидает выделения процессора.

Running (Выполняется) — процесс в текущий момент выполняется на процессоре.

Waiting (Ожидание) — процесс ожидает завершения какого-либо события (например, операции ввода-вывода).

Terminated (Завершён) — процесс завершил выполнение, и его ресурсы подлежат освобождению.

Переходы между состояниями

Процессы могут переходить из одного состояния в другое в зависимости от событий и действий операционной системы:

New → Ready: процесс создан и загружен в память.

Ready → Running: планировщик операционной системы назначает процессу центральный процессор.

Running → Waiting: процесс делает запрос на операцию, требующую ожидания (например, чтение файла)

Waiting → Ready: запрошенное событие завершено, процесс может быть выполнен.

Running → Ready: прерывание таймера или вытеснение другим процессом.

Running → Terminated: выполнение завершено, либо возникла ошибка.

Блок управления процессом (Process Control Block, PCB)

Каждому процессу сопоставляется специальная структура данных — блок управления процессом (PCB), содержащая:

уникальный идентификатор процесса (PID);

текущее состояние процесса;

значения регистров процессора;

информацию о планировании (приоритет, квота времени и т. д.);

информацию о размещении в памяти;

сведения об открытых файлах и используемых ресурсах.

PCB используется операционной системой для управления процессами и организации многозадачности.

Потоки и процессы

Один процесс может содержать один или несколько потоков исполнения. Поток — это минимальная единица выполнения, обладающая собственным стеком и регистровым состоянием, но использующая общее адресное пространство и ресурсы процесса. Многопоточность позволяет выполнять несколько операций параллельно внутри одного процесса.

Планирование процессов в системах пакетной обработки — это один из ключевых аспектов управления процессами в операционных системах, ориентированных на выполнение задач без вмешательства пользователя во время исполнения. Ниже приведено подробное объяснение.

Планирование процессов в системах пакетной обработки

1. Общие сведения о пакетной обработке

Системы пакетной обработки (batch processing systems) предназначены для выполнения серий (пакетов) заданий без непосредственного взаимодействия с пользователем. Пользователь подготавливает задания (job), которые включают:

саму программу;

входные данные;

команды управления (job control language — JCL).

Задания поступают в очередь и выполняются последовательно или согласно определённому алгоритму.

2. Цели планирования в пакетных системах

Планирование процессов в таких системах направлено на:

максимиацию пропускной способности (числа заданий, выполненных в единицу времени);

минимизацию времени отклика (времени от подачи задания до начала его выполнения);

эффективное использование ресурсов (процессора, памяти, ввода-вывода);

упорядоченное выполнение задач с учётом приоритетов.

3. Алгоритмы планирования в пакетных системах

Наиболее распространённые алгоритмы планирования в системах пакетной обработки:

1. FCFS (First-Come, First-Served) — "первым пришёл — первым обслужен"

Задания обрабатываются в порядке поступления.

Простой в реализации, но может приводить к эффекту голодания коротких задач из-за длительного выполнения предыдущих.

2. SJF (Shortest Job First) — "сначала короткие задания"

Задания с наименьшим временем выполнения выполняются первыми.

Оптимален по среднему времени ожидания, но требует знания времени выполнения, что не всегда возможно.

Возможна задержка длинных заданий (голодание).

4. Приоритетное планирование (Priority Scheduling)

Каждому заданию присваивается приоритет.

Выполняются задания с более высоким приоритетом.

Возможны проблемы с голоданием низкоприоритетных задач (можно использовать возраст приоритета — "aging").

4. Особенности реализации

В пакетных системах отсутствует интерактивность, поэтому ключевым является оптимизация общего времени обработки заданий.

Часто используется пакетная очередь, где задания могут быть отсортированы заранее, в зависимости от заданных критериев.

В некоторых случаях применяется мультипрограммирование, чтобы использовать время ввода-вывода эффективно (выполняется одно задание, пока другое ждёт ввода-вывода).

5. Пример работы планировщика

Допустим, в систему поступают три задания:

Задание Время выполнения (мин) Время поступления

A 5 0

B 2 1

C 8 2

FCFS: порядок A → B → C.

SJF: порядок B → A → C (минимальное время выполнения).

HRRN: зависит от времени ожидания и длины задания.

6. Вывод

Планирование процессов в системах пакетной обработки играет ключевую роль в эффективности использования вычислительных ресурсов. Выбор алгоритма зависит от целей: максимальной загрузки процессора, минимизации времени ожидания, предотвращения голодания и т. д. В современных системах пакетная обработка сохраняет актуальность, особенно при выполнении массовых вычислений, рендеринга, обработки логов и других задач без интерактивного интерфейса.