9. Понятие процесса. Состояния процесса

Процесс — это программа, находящаяся в состоянии выполнения. Каждый процесс имеет:

собственное адресное пространство,

регистры процессора,

стек и кучу,

дескриптор процесса (PID).

Состояния процесса (классическая модель):

Новый (New) — процесс создан, но ещё не запущен.

Готовность (Ready) — процесс ожидает выделения CPU.

Выполнение (Running) — процесс выполняется на CPU.

Ожидание (Waiting/Blocked) — процесс ждёт внешнего события (ввод/вывод, сигнал).

Завершённый (Terminated) — процесс завершил выполнение.

(Возможны дополнительные состояния, например, "Приостановлен" (Suspended) в многозадачных системах.)

10. Планирование процессов в системах пакетной обработки

Пакетная обработка — выполнение задач без интерактивного взаимодействия (например, обработка данных, компиляция).

Алгоритмы планирования:

FCFS (First-Come, First-Served) — в порядке очереди.

Плюсы: простота.

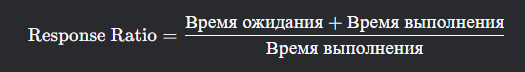
Минусы: возможны длительные простои из-за "долгих" задач.

SJF (Shortest Job First) — сначала выполняются самые короткие задачи.

Плюсы: минимизирует среднее время ожидания.

Минусы: требует знания времени выполнения.

HRRN (Highest Response Ratio Next) — выбирает задачу с наибольшим коэффициентом:



(В пакетных системах важна максимальная загрузка CPU, а не отзывчивость.)

11. Планирование процессов в интерактивных системах

Интерактивные системы требуют быстрого отклика (например, ОС для ПК).

Алгоритмы планирования:

Round Robin (RR) — циклическое выделение квантов времени (например, 10-100 мс).

Плюсы: справедливость, низкая задержка.

Минусы: возможны частые переключения контекста.

Multilevel Queue (MLQ) — несколько очередей с разными приоритетами (например, системные процессы → интерактивные → фоновые).

Multilevel Feedback Queue (MLFQ) — динамическое изменение приоритета (если процесс использует весь квант — понижается приоритет).

(Критерий эффективности — время отклика.)

12. Планирование процессов в системах реального времени

Системы реального времени (RTOS) требуют гарантированного времени выполнения.

Типы:

Жёсткие (Hard RT) — нарушение сроков недопустимо (управление роботом).

Мягкие (Soft RT) — небольшие задержки допустимы (стриминг).

Алгоритмы:

Rate-Monotonic (RM) — статический приоритет (чем чаще задача, тем выше приоритет).

Earliest Deadline First (EDF) — динамический приоритет (ближайший дедлайн выполняется первым).

(Критерий — предсказуемость, а не fairness.)

13. Операции над процессами

Создание (fork() в Unix, CreateProcess() в Windows).

Завершение (exit()).

Ожидание (wait() — родительский процесс ждёт завершения дочернего).

Замена образа процесса (exec() — загрузка новой программы).

Изменение приоритета (nice в Unix).

Сигналы (kill — отправка сигналов процессу).

14. Приоритеты процессов

Приоритет определяет порядок доступа к CPU.

Типы:

Статический (назначается при создании).

Динамический (меняется в runtime, например, в MLFQ).

Примеры:

В Windows: 32 уровня (0-31).

В Unix: nice от -20 (высокий) до +19 (низкий).

(Высокоприоритетные процессы могут вызывать инверсию приоритетов.)

15. Понятие потока. Ресурсы и потоки в ОС

Поток (thread) — легковесный процесс, разделяющий память с другими потоками.

Ресурсы процесса:

Общие: память, файлы.

Индивидуальные: регистры, стек.

Плюсы потоков:

Быстрое создание/переключение.

Эффективное использование CPU (параллелизм).

(В ОС потоки могут быть: пользовательскими (управляются библиотекой) или ядерными (управляются ОС).)

16. Понятие ресурса. Классификация ресурсов

Ресурс — любой компонент системы (CPU, память, устройства ввода-вывода).

Классификация:

По типу:

Аппаратные (CPU, RAM, HDD).

Программные (файлы, сокеты, мьютексы).

По возможности разделения:

Разделяемые (RAM, файлы).

Неразделяемые (принтер).

По способу выделения:

Статические (выделяются на всё время).

Динамические (запрашиваются при необходимости).

17. Дисциплины распределения ресурсов на основе очередей

FIFO (First-In, First-Out) — первым пришёл, первым обслужен.

LIFO (Last-In, First-Out) — стековая модель.

Приоритетные очереди — высокоприоритетные задачи обслуживаются первыми.

Round Robin — циклическое распределение.

(Используются в планировщиках, диспетчерах ввода-вывода и сетевых очередях.)

18. Основные элементы графических интерфейсов

Окна (Windows, Tabs).

Меню (File, Edit).

Кнопки (Button).

Поля ввода (Textbox).

Полосы прокрутки (Scrollbar).

Диалоговые окна (Modal/Non-modal).

Панели инструментов (Toolbar).

(Основаны на событийно-ориентированной модели.)

19. Интерфейс прикладного программирования (API)

API — набор функций, предоставляемых ОС или библиотекой для взаимодействия с системой.

Примеры API в ОС:

POSIX (Unix/Linux).

WinAPI (Windows).

Java API (JVM).

Типы API:

Системные вызовы (read(), write()).

Библиотечные функции (стандартная библиотека C).

Сетевые API (HTTP-запросы).

(API упрощают разработку, абстрагируя низкоуровневые детали.)