**1.Понятие процесса. Состояния процесса**

**1.1. Процесс и его определение**

**Процесс** — это программа в состоянии выполнения. Он включает:

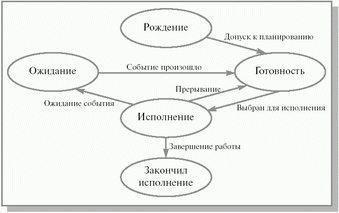
* Код программы.
* Данные (переменные, буферы).
* Состояние процессора (регистры, счетчик команд).
* Системные ресурсы (открытые файлы, сетевые соединения).

**1.2. Состояния процесса**

Процесс может находиться в одном из следующих состояний:

| **Состояние** | **Описание** |
| --- | --- |
| **Новый (New)** | Процесс создан, но еще не загружен в память. |
| **Готовый (Ready)** | Процесс загружен в ОЗУ и ожидает выделения процессорного времени. |
| **Исполняемый (Running)** | Процесс выполняется на CPU. |
| **Ожидание (Waiting)** | Процесс приостановлен (ждет завершения I/O, сигнала или ресурса). |
| **Завершенный (Terminated)** | Процесс завершил выполнение или был принудительно остановлен. |

**Диаграмма состояний процесса:**



**2. Планирование процессов**

**2.1. Планирование в системах пакетной обработки**

**Примеры:** мейнфреймы, научные вычисления.

**Алгоритмы:**

* **FCFS (First-Come, First-Served)** – процессы выполняются в порядке поступления.
* **SJF (Shortest Job First)** – приоритет отдается самым коротким задачам.
* **HRRN (Highest Response Ratio Next)** – выбирается процесс с наилучшим соотношением времени ожидания и времени выполнения.

**2.2. Планирование в интерактивных системах**

**Примеры:** Windows, Linux, macOS.

**Алгоритмы:**

* **Round Robin (RR)** – каждому процессу выделяется квант времени (например, 10 мс).
* **Multilevel Queue** – процессы делятся на группы (например, системные и пользовательские).
* **Multilevel Feedback Queue** – динамическое изменение приоритетов.

**2.3. Планирование в системах реального времени**

**Примеры:** управление роботами, медицинские системы.

**Требования:**

* **Жесткое реальное время** – гарантированное выполнение в срок.
* **Мягкое реальное время** – желательно, но не критично.

**Алгоритмы:**

* **Rate-Monotonic (RM)** – более высокий приоритет у задач с меньшим периодом.
* **Earliest Deadline First (EDF)** – выполняется задача с ближайшим дедлайном.

**3. Операции над процессами**

| **Операция** | **Описание** |
| --- | --- |
| **Создание** | fork() (Unix), CreateProcess() (Windows). |
| **Завершение** | exit(), kill(). |
| **Ожидание** | wait() – родительский процесс ждет завершения дочернего. |
| **Замена образа** | exec() – замена кода процесса другой программой. |
| **Синхронизация** | Семафоры, мьютексы, мониторы. |

**4. Приоритеты процессов**

Приоритет определяет очередность выполнения:

* **Статический** – назначается при создании (например, системные процессы имеют высокий приоритет).
* **Динамический** – меняется во время выполнения (например, в Multilevel Feedback Queue).

**Примеры:**

* В Linux: nice (от -20 до 19, где -20 – наивысший приоритет).
* В Windows: SetPriorityClass().

**5. Понятие потока. Ресурсы и потоки**

**5.1. Поток (Thread)**

**Поток** — это легковесный процесс, выполняемый в рамках одного адресного пространства.

**Отличие от процесса:**

| **Критерий** | **Процесс** | **Поток** |
| --- | --- | --- |
| **Ресурсы** | Изолированная память | Общая память с другими потоками |
| **Создание** | Медленное (требует выделения памяти) | Быстрое |
| **Переключение** | Дорогое (сохранение контекста) | Дешевое |

**5.2. Ресурсы в ОС**

Ресурсы делятся на:

1. **Аппаратные** (CPU, память, диски).
2. **Программные** (файлы, сокеты, семафоры).

**6. Классификация ресурсов**

| **Критерий** | **Типы ресурсов** |
| --- | --- |
| **По типу доступа** | Разделяемые (память), монопольные (принтер). |
| **По управлению** | Выделяемые (динамически) и статические (фиксированные). |
| **По потреблению** | Возобновляемые (память) и невозобновляемые (сообщения). |

**7. Распределение ресурсов на основе очередей**

**7.1. FIFO (First-In, First-Out)**

* Процессы обслуживаются в порядке поступления.
* Проблема: "голодание" длинных задач.

**7.2. Приоритетные очереди**

* Системные процессы получают ресурсы первыми.

**7.3. Round-Robin для ресурсов**

* Каждый процесс получает ресурс на фиксированное время.

**8. Графические интерфейсы (GUI) и API**

**8.1. Основные элементы GUI**

* **Окна** – области отображения информации.
* **Кнопки** – элементы управления.
* **Меню** – списки команд.
* **Виджеты** – текстовые поля, ползунки, чекбоксы.

**8.2. Интерфейс прикладного программирования (API)**

**API** — это набор функций для взаимодействия программ с ОС или библиотеками.

**Примеры:**

* **WinAPI** – для Windows.
* **POSIX** – для Unix-систем.
* **OpenGL** – для графики.