9. Понятие процесса. Состояния процесса.

Понятие процесса:

Процесс — это программа, находящаяся в состоянии выполнения. Он представляет собой единицу работы, которой операционная система (ОС) выделяет ресурсы: процессорное время, память, дескрипторы файлов и другие системные ресурсы.

Каждый процесс имеет:

1. собственное пространство адресов (код, данные, стек),
2. уникальный идентификатор процесса (PID),
3. контекст выполнения (регистры процессора, счётчик команд и т. д.),
4. нформацию о состоянии.

Процесс может быть как пользовательским (например, браузер, текстовый редактор), так и системным (службы ОС, демоны).

Состояния процесса:

В процессе своей жизни любой процесс переходит через несколько состояний:

| Состояние | Описание |
| --- | --- |
| Создание (new) | Процесс создаётся. Операционная система выделяет ему ресурсы. |
| Готовность (ready) | Процесс готов к выполнению и ждёт, пока ОС выделит ему процессор. |
| Выполнение (running) | Процесс выполняется — ему выделено время процессора. |
| Ожидание (waiting/block) | Процесс приостановлен и ждёт завершения внешнего события (например, чтения файла). |
| Завершение (terminated/exit) | Процесс завершил работу или был завершён ОС или пользователем. |

10. Планирование процессов в системах пакетной обработки

Что такое система пакетной обработки?

Система пакетной обработки (batch processing) — это тип операционной системы, в которой задания (процессы) выполняются без взаимодействия с пользователем. Программы группируются в пакеты, каждый из которых выполняется последовательно.

1. FCFS (First-Come, First-Served) — «Первым пришёл — первым обслужен»

* Задания обрабатываются в порядке поступления.

2. SJF (Shortest Job First) — «Сначала короткое задание»

* Сначала выполняется задание с наименьшим временем выполнения.

#### **3. Приоритетное планирование (Priority Scheduling)**

* Каждому заданию назначается **приоритет**.

11. Планирование процессов в интерактивных системах

Что такое интерактивная система?

Интерактивная система — это операционная система, в которой пользователь активно взаимодействует с программами во время их выполнения.

В интерактивных системах важно быстро реагировать на действия пользователя, поэтому планирование процессов должно обеспечивать отзывчивость, справедливость и равномерное распределение ресурсов.

Основные цели планирования в интерактивных системах:

* Минимизировать время отклика (response time).
* Обеспечить справедливый доступ к процессору.
* Избегать голодания процессов.
* Эффективно управлять переключениями контекста.

Основные алгоритмы планирования в интерактивных системах:

Round Robin (RR) — циклический (карусельный) алгоритм

* Каждому процессу выделяется тайм-квант (например, 100 мс).

#### **Многоуровневое планирование с обратной связью (Multilevel Feedback Queue, MLFQ)**

* Несколько очередей с **разными приоритетами и квантами времени**.

#### **Приоритетное планирование с вытеснением**

* Процесс с более высоким приоритетом **вытесняет** текущий.

12. Планирование процессов в системах реального времени

Что такое система реального времени?

Система реального времени (Real-Time System) — это система, в которой время отклика на событие критично, и результат должен быть получен в строго заданные временные рамки.

Классификация систем реального времени:

| Тип системы | Описание |
| --- | --- |
| Жёсткая (hard real-time) | Пропуск крайнего срока недопустим (например, запуск подушки безопасности). |
| Мягкая (soft real-time) | Нарушение сроков допустимо, но снижает качество (например, видеоконференция). |
| Статическая/динамическая | В статической известно всё о задачах заранее, в динамической — задачи приходят во время исполнения. |

Основные алгоритмы планирования в системах реального времени:

1. Rate Monotonic Scheduling (RMS)

* Статический приоритет: чем короче период задачи, тем выше её приоритет.

#### **2. Earliest Deadline First (EDF)**

* **Динамический приоритет**: приоритет выше у задачи с **ближайшим дедлайном**.

#### **3. Least Laxity First (LLF)**

* Выбирается задача с **наименьшим «запасом по времени»**:  
  **Laxity = Deadline − (Текущее время + Оставшееся время выполнения)**.

13. Операции над процессами

В операционных системах процесс — это программа в состоянии выполнения. Для управления процессами ОС предоставляет различные операции, обеспечивающие создание, завершение, приостановку и взаимодействие между процессами.

Основные операции над процессами:

1. Создание процесса (Create / fork / spawn)

2. Завершение процесса (Exit / terminate / kill)

3. Ожидание завершения другого процесса (Wait)

14. Приоритеты процессов

Что такое приоритет процесса?

Приоритет процесса — это числовое значение, определяющее, насколько важным операционная система считает процесс при выборе следующего для выполнения.

Чем выше приоритет, тем чаще и раньше планировщик ОС выбирает данный процесс для выполнения.

Зачем нужны приоритеты?

* Чтобы более важные задачи получали больше процессорного времени (например, системные службы).
* Чтобы реализовать многозадачность с разным уровнем срочности.
* Чтобы управлять поведением фоновых и активных приложений.

Виды приоритетов:

| Тип приоритета | Описание |
| --- | --- |
| Статический | Назначается при создании процесса и не меняется. Применяется в системах реального времени. |
| Динамический | Может изменяться ОС во время выполнения (например, если процесс долго не работает — повышается). Используется в универсальных ОС. |
| Относительный | Приоритет в пределах пользовательского пространства (например, nice в Linux). |
| Абсолютный | Задаётся в пределах всей системы (например, kernel-level приоритет в Windows). |

15. Понятие потока. Ресурсы и потоки в операционной системе

Что такое поток (thread)?

Поток (или нить) — это наименьшая единица выполнения внутри процесса.  
Один процесс может содержать один или несколько потоков, которые разделяют между собой память и ресурсы процесса, но выполняются независимо.

Отличия процесса и потока:

| Характеристика | Процесс | Поток |
| --- | --- | --- |
| Память | Своя (изолирована) | Общая с другими потоками |
| Управление | Тяжёлое (создание, переключение) | Лёгкое и быстрое |
| Связь с другими | Через IPC (межпроцессное взаимодействие) | Через общую память |
| Пример использования | Запуск программ, системные службы | Параллельная работа внутри программы |

#### Ресурсы и потоки

#### Процесс включает: **Код (текст программы), Данные, Открытые файлы, Окружение, Системные ресурсы (дескрипторы, сокеты и т.п.)**

#### Поток включает: **Свой стек, Счётчик команд (Program Counter), Регистры процессора, Собственный контекст выполнения**

16. Понятие ресурса. Классификация ресурсов

Что такое ресурс?

Ресурс в операционной системе — это любая сущность, необходимая процессу (или потоку) для выполнения. Ресурсы могут быть как физическими (например, процессор, память), так и логическими (файлы, семафоры и т.п.).

Классификация ресурсов:

1)По происхождения

- физические

- логические

2)По способу использования

-разделяемые

-неразделяемые

3)По длительности использования

-временные

-постоянные

4)По структуре

-дискретные

-непрерывные

17. Дисциплины распределения ресурсов на основе очередей

Что это такое?

Дисциплина распределения ресурсов — это метод (алгоритм), по которому операционная система решает, в каком порядке процессы получают доступ к ограниченным ресурсам.

Когда ресурс занят, остальные процессы становятся в очередь ожидания, и ОС должна решить, кому выдать ресурс первым, когда он освободится.

Основные дисциплины распределения на основе очередей

| № | Дисциплина | Описание |
| --- | --- | --- |
| 1 | FIFO (First-In, First-Out)Очередь по времени поступления | Кто раньше стал в очередь — тот и получает ресурс первым. |
| 2 | Приоритетная очередь(Priority Queue) | Процессы с высшим приоритетом обслуживаются первыми, даже если поступили позже. |
| 3 | Круговой (Round Robin) | Каждому процессу даётся фиксированное квантовое время, по истечении которого он уступает место следующему. |
| 4 | LIFO (Last-In, First-Out) | Последний вошедший обслуживается первым. Используется редко — может привести к голоданию. |
| 5 | Случайный выбор (Random) | Ресурс передаётся случайному процессу из очереди. Подходит для симуляций и игр. |
| 6 | На основе справедливости (Fair Queueing) | Система пытается распределить ресурсы равномерно между всеми процессами. |
| 7 | Многоуровневая очередь (Multilevel Queue) | Существует несколько очередей по приоритетам, и между ними действует отдельный алгоритм. |

18. Основные элементы графических интерфейсов (GUI)

Графический интерфейс пользователя (GUI) — это способ взаимодействия с программой с помощью визуальных компонентов: окон, кнопок, полей ввода и т. д. GUI позволяет пользователю управлять приложением интуитивно, без необходимости запоминать команды.

Основные элементы GUI:

| Элемент GUI | Описание |
| --- | --- |
| Окно (Window) | Основной контейнер, в котором располагаются другие элементы интерфейса. Обычно имеет заголовок, кнопки "свернуть", "развернуть", "закрыть". |
| Кнопка (Button) | Элемент для запуска действия. Пример: кнопка "Сохранить", "ОК", "Отмена". |
| Поле ввода (TextBox/Input Field) | Позволяет пользователю вводить текст. Может быть однострочным (логин) или многострочным (комментарий). |
| Метка (Label) | Отображает текстовую информацию, часто используется как подпись к другим элементам. |
| Флажок (Checkbox) | Элемент для выбора одной или нескольких независимых опций. Пример: Подписаться на рассылку. |
| Переключатель (Radio Button) | Позволяет выбрать одну опцию из группы. Пример: ○ Мужской ○ Женский. |
| Выпадающий список (ComboBox) | Список, из которого можно выбрать один вариант. Экономит место при множестве опций. |
| Список (ListBox) | Показывает список элементов, из которых можно выбрать один или несколько. |
| Полоса прокрутки (Scroll Bar) | Позволяет просматривать контент, не помещающийся в окне (горизонтально/вертикально). |
| Панель инструментов (Toolbar) | Набор иконок или кнопок для быстрого доступа к функциям. |
| Строка состояния (Status Bar) | Показывает информацию о текущем состоянии программы. Например: "Сохранено", "Подключение установлено". |
| Диалоговое окно (Dialog Box) | Всплывающее окно для подтверждения, ввода информации или уведомления. Пример: окно "Сохранить изменения перед выходом?". |
| Изображения и иконки | Используются для визуального представления объектов (например, значок корзины, папки, файла). |

18. Интерфейс прикладного программирования (API)

Что такое API?

API (Application Programming Interface) — это набор правил, протоколов и инструментов, которые позволяют разным программным приложениям взаимодействовать друг с другом.

API определяет, каким образом одна программа может использовать функции или данные другой программы или сервиса без необходимости знать внутреннее устройство этой программы.

Основные аспекты API:

* Интерфейс — это набор доступных функций, методов, классов или команд.
* Протокол — правила обмена сообщениями между программами.
* Формат данных — стандарты представления информации (например, JSON, XML).

Виды API:

| Вид API | Описание |
| --- | --- |
| Локальный API | Вызовы функций внутри одной системы или приложения. |
| Веб-API (REST, SOAP) | API, предоставляемые через интернет, для доступа к сервисам. |
| Библиотечный API | Интерфейс к библиотекам программного обеспечения. |