Обзор подходов к созданию дочерних процессов и разделению ресурсов дочерним и родительским процессами в операционных системах

Операционные системы

Сахно Алёна Юрьевна

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

Студ.билет 1132243813

НКАбд-04-24

Информация

Преподователь

- Кулябов Дмитрий Сергеевич
- д.ф.-м.н., профессор
- профессор кафедры прикладной информатики и теории вероятностей
- Российский университет дружбы народов
- kulyabov-ds@rudn.ru
- https://yamadharma.github.io/ru/

Вводная часть

Вводная часть

В современных операционных системах управление процессами является одной из ключевых задач, обеспечивающих эффективное использование ресурсов и выполнение многозадачных приложений. Создание дочерних процессов позволяет приложениям выполнять несколько задач одновременно, улучшая производительность и отзывчивость.

Разделение ресурсов между родительским и дочерним процессами — важный аспект многозадачности. Оно определяет, какие ресурсы могут быть использованы совместно, а какие должны оставаться изолированными. Это влияет на производительность системы, безопасность и стабильность приложений.

Актуальность

- Многозадачность и производительность
- Безопасность и изоляция
- Разработка распределенных систем
- Современные парадигмы программирования

Объект и предмет исследования

- Объектом исследования являются операционные системы, а именно их механизмы управления процессами.
- Предметом исследования являются подходы и методы создания дочерних процессов и разделения ресурсов между родительским и дочерним процессами.

Цель работы

- Показать обзор подходов к созданию дочерних процессов и разделению ресурсов дочерним и родительским процессами в операционных системах.

Задачи

- Объяснить концепцию процессов и потоков
- Изучить механизм создания дочерних процесссов
- Провести сравнение различных операционных систем

- Процесс - это экземпляр программы во время выполнения, независимый объект, которому выделены системные ресурсы -Поток - объекты в составе процесса, которые отвечают за выполнение кода.

- Примеры использования потоков в реальных приложениях

Подход создания дочерних процесов

Дочерний процесс это новый процесс, созданный уже существующим процессом, который мы называем родительским.

fork()

Часто задаваемые вопросы

- Что произойдет, если родительский процесс завершится?
- Можно ли создать дочерний процесс без использования fork()?
- Как узнать количество дочерних процессов у определенного процесса
- Что такое орфанные процессы?
- Как предотвратить создание дочерних процессов?

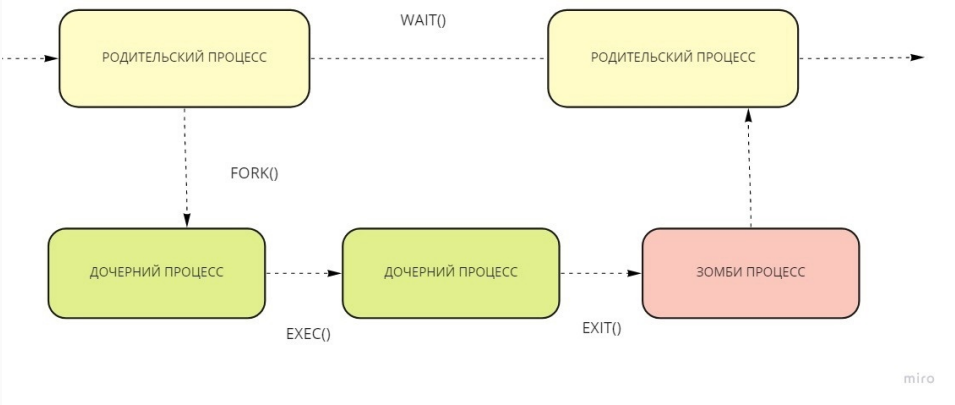
Разделение ресурсов между родительским и дочерним процессами

Unix/Linux - Память При использовании fork() создается копия адресного пространства родитель- ского процесса. - Файлы и дескрипторы Оба процесса могут наследовать открытые файловые дескрипторы от родителя - Сигналы Дочерние процессы могут получать сигналы от родительских процессов для управления выполнением

Разделение ресурсов между родительским и дочерним процессами

Windows - Память В Windows каждый процесс имеет собственное адресное пространство. - Дескрипторы Дочерние процессы могут наследовать дескрипторы, но необходимо явно указывать это при создании процесса. - IPC (межпроцессное взаимодействие) Windows предоставляет различные механизмы для IPC, такие как именованные каналы (named pipes), очереди сообщений и общая память (shared memory)

Разделение ресурсов между родительским и дочерним процессами



Изоляция ресурсов

- Контейнеры и виртуализация
- Системы контроля доступа
- Потоки и легковесные процессы
- Завершение процессов

Заключение

Создание дочерних процессов и управление ресурсами между ними являются важными аспектами работы операционных систем. Различные подходы и ме- ханизмы обеспечивают гибкость в создании и управлении процессами, а также позволяют эффективно использовать ресурсы системы.

Вывод

я рассказала про обзор подходов к созданию дочерних процессов и разделению ресурсов дочерним и родительским процессами в операционных системах.