

Улучшение средств синхронизации в ОСРВ Embox

Курсовая работа

Научный руководитель

АНТОН КОЗЛОВ

асп. каф. Системного Программирования

Докладчик

Вита Логинова

студент 444 гр.

Многозадачность: преимущества

- Необходимое условие для современных систем
- Может сильно ускорить программу
- Распределение ресурсов процессора

Многозадачность: проблемы

- Сложность в разработке
 - Неэффективная реализация может сильно замедлить программу
 - Надежность
 - Защита памяти
 - Обработка прерываний
 - Предотвращение ситуации гонки, инверсии приоритета...

Методы IPC

- обмен сообщениями
- синхронизация
- разделяемая память
- удаленные вызовы

Задача

- Исследовать различные методы IPC
- Усовершенствовать существующую реализацию средств синхронизации в ядре ОС Embox
- Оценить полученную реализацию

Содержание работы

Пользовательское приложение

Библиотека IPC

Ядро ОС

Механизм ожидания

Планировщик

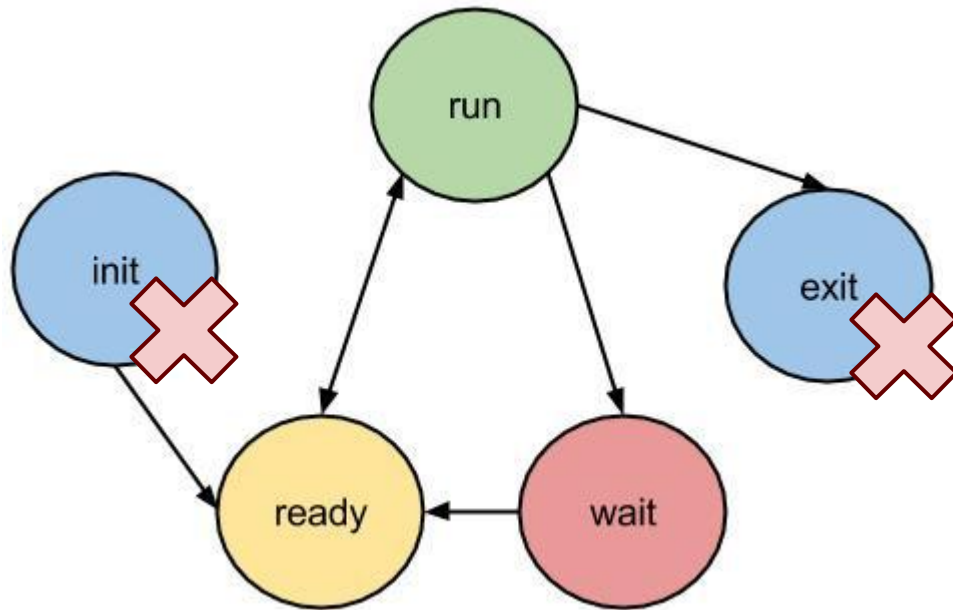
Прерывания

Существующие примитивы

- mutex
- cond
- semaphore
- ...

Проблема: для предотвращения тупика необходимо отключать планировщик

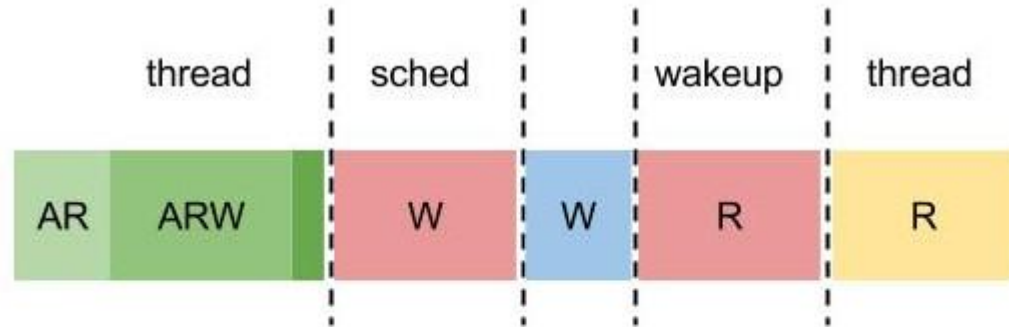
Машина состояний потока



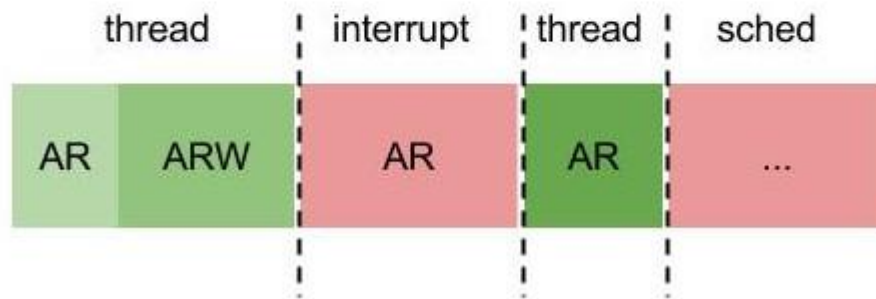
Промежуточные состояния:
Перед `schedule()` поток уже помечается как
ждущий

Машина состояний

1) Событие случилось **после** того, как поток передал управление планировщику



2) Событие случилось **до** того, как поток передал управление планировщику



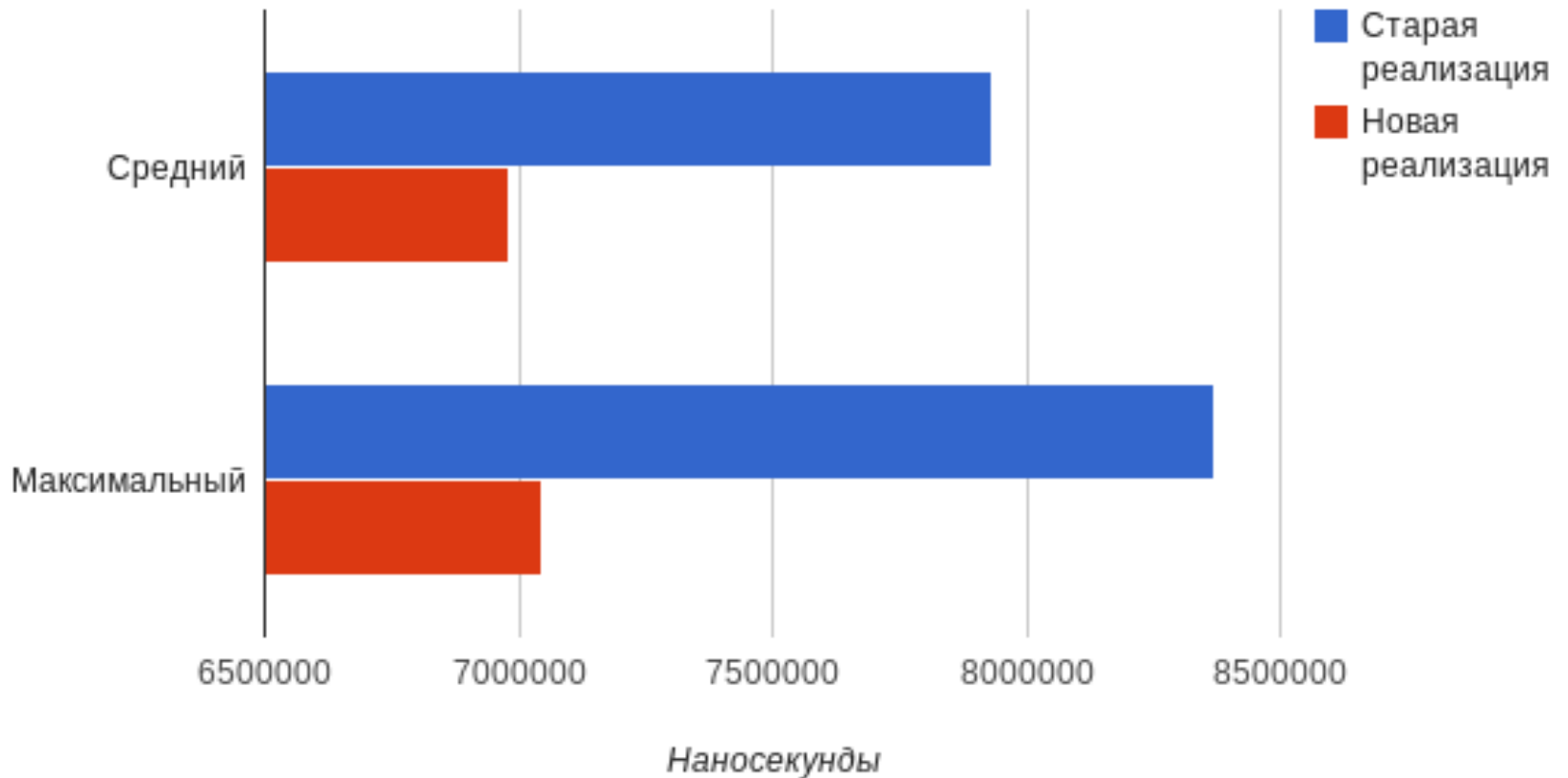
A = active; R = ready; W = waiting

Новый механизм ожидания

```
waitq_wait_prepare(waitq, waiq_link);  
sched_wait_timeout(timeout);  
waitq_wait_cleanup(waitq, waiq_link);
```

Сравнение реализаций

Тест с мьютексами на 20 потоков



Результаты

- Исследованы примитивы синхронизации в ОС Embox
- Исследована текущая реализация многопоточности в ядре ОС Embox
- Переработана машина состояний планировщика
- Переработан механизм ожидания событий
- Примитивы синхронизации переработаны согласно новому интерфейсу
- Проведено сравнение реализаций