

ВСТРОЕННЫЕ ИНФОРМАЦИОННО- УПРАВЛЯЮЩИЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ

*Операционные системы
реального времени*

Кафедра АСВК,
Лаборатория Вычислительных Комплексов

Содержание лекции

- Требования к ОС РВ
- Понятие процесса и потока (thread)
- Планирование процессов с приоритетами
- Взаимодействие процессов в ОС РВ
- Служба времени ОС РВ
- Принципы построения ОС РВ.
- Примеры ОС РВ.

Основные требования к ОС РВ

Предсказуемость реакции и многозадачность

- Наличие механизма приоритетов процессов и вытеснения (preemption)
- Предсказуемые механизмы синхронизации и взаимодействия
- Предсказуемые задержки
- Малые задержки

Основные виды задержек

- Переключение контекста
 - Реакция на прерывание
 - Выполнение системного вызова
 - Начальная загрузка системы
-
- Потоки (threads) вместо процессов – как можно меньший контекст
 - работа без виртуальной памяти (без преобразования адресов)

Основные требования к ОС (2)

- Минимальность по используемой памяти
- Загрузка из ПЗУ
- Отсутствие дисковой памяти
- Настраиваемость на аппаратуру (Пакет поддержки модуля – BSP)
- Соответствие отраслевым стандартам (POSIX 1003.1a,b,c; ARINC-653, и т.д.)

Соответствие требованиям по безопасности (DO-178, ...)

Процессы и потоки (threads)

- Исполняемый файл – тело процесса
- Функция-тело потока
- Параметры планирования
 - Дисциплина планирования
 - приоритет
 - Признак наследования от родителя
- контекст (стек, куча, дескрипторы, ...)

Приоритетное планирование (встроенный планировщик ОС)

- Имеется диапазон значений приоритетов (256 для ОС-2000)
- Для каждого приоритета – своя очередь
- Управление получает поток в голове очереди с наивысшим приоритетом
- Появился готовый поток с большим приоритетом => вытеснение
- SCHED_RR => перепланир по истечении кванта времени (карусель, round-robin)
- sched_yield() – явная отдача управления

Планирование периодических задач

- Дано:
 - Набор задач $T_1, T_2, \dots, T_n$
 - Период $P_i, i=1..n$
 - Директивный срок D_i
 - Время выполнения C_i
 - $C_i \leq D_i \leq P_i$
- Требуется:
 - Проверить возможность планирования без нарушения директивных сроков
 - Назначить приоритеты потокам

Планирование Rate Monotonic

- Каждой задаче – своё значение приоритета (неизменное)
- Наибольший приоритет – задаче с наименьшим периодом
- Достаточное условие планируемости

$$U \leq n(2^{1/n} - 1)$$

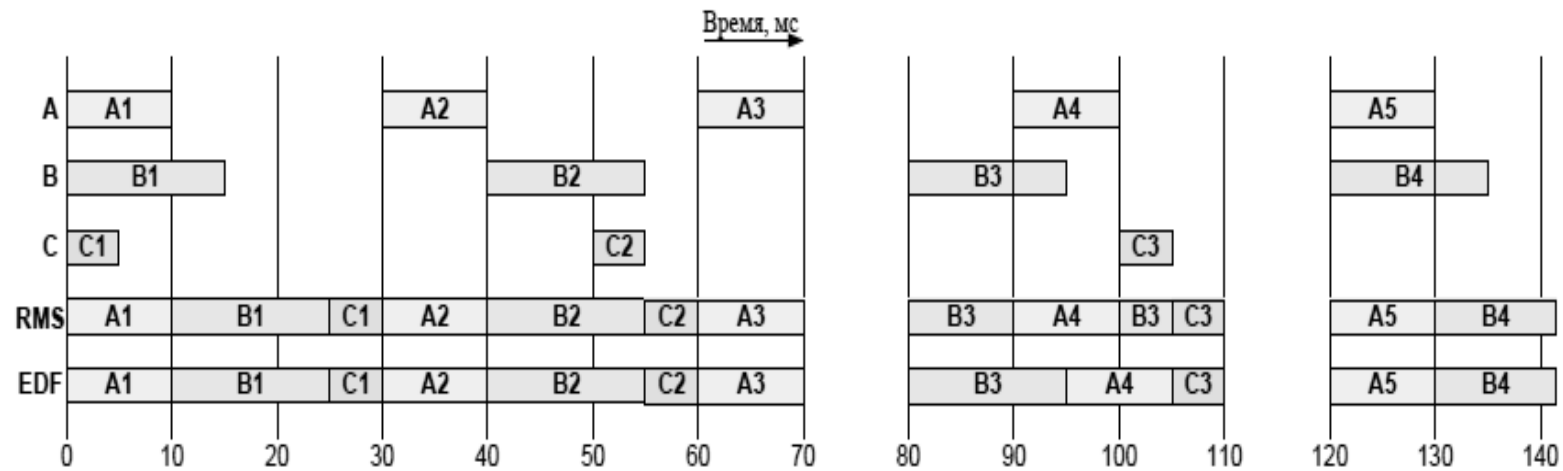
$$U = \sum(U_i), U_i = C_i/P_i;$$

Планирование EDF (Earliest Deadline First)

- Меньший текущий директивный срок => больший приоритет
- Необходимое и достаточное условие планируемости:

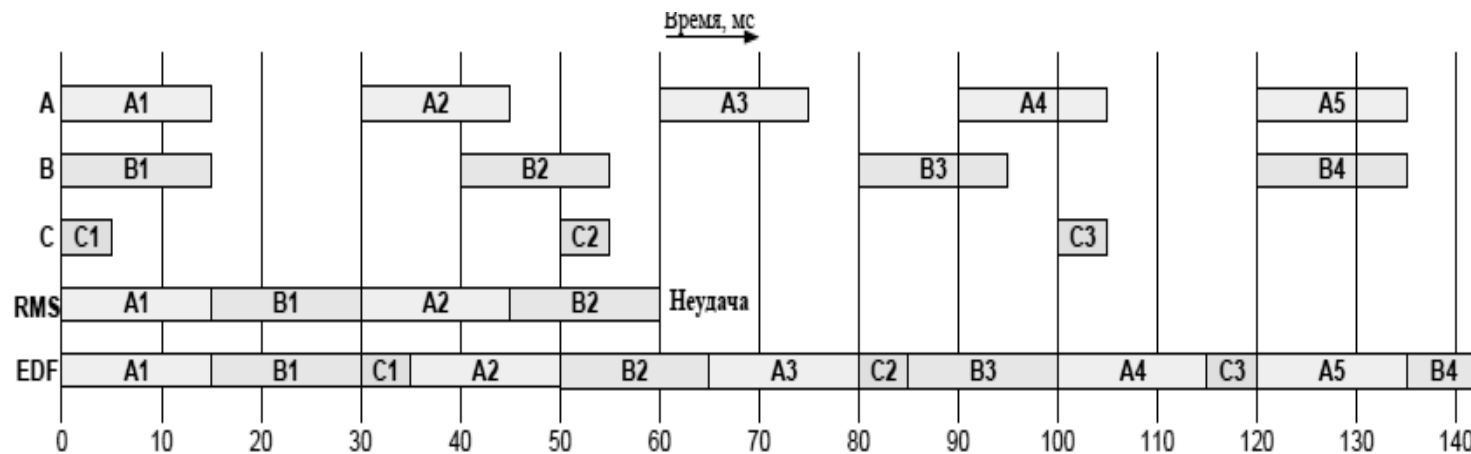
$$U < 1$$

Планирование: пример



A: 10 мс / 30 мс; B: 15/40; C: 5/50

Планирование: пример (2)



Здесь для A: 15 / 40 мс

Справляется только алгоритм EDF

Взаимодействие процессов (потоков)

Очереди сообщений

- `mq_open()` // имя очереди `const char*`
- `mq_close()`
- `mq_send()` // длина сообщения, приоритет
- `mq_receive()`
 - В зависимости от параметров очереди – либо ожидание на пустой очереди, либо ошибка
- `mq_notify()` // установить сигнал при записи в пустую очередь

Семафоры

Аналогично Unix без реального времени

`sem_open()`, `sem_close()`

`sem_post()`

`sem_wait()`, `sem_trywait()`

Очередь к семафору – по приоритетам
потоков

Мьютексы

- `#include <pthread.h>`
- `pthread_mutex_init()`
- `pthread_mutex_lock()`
- `pthread_mutex_trylock()`
- Освободить mutex может только захвативший его поток
- «Упрощённый семафор»

Сигналы реального времени

По сравнению с «обычными» сигналами:

- расширенный диапазон;
- постановка в очередь;
- расширенные возможности задания реакции (маска блокировки при обработке сигнала)

Разделяемая память

- общие сегменты памяти
- Файлы, отображённые на память

Представление времени

```
#include <time.h>
struct timespec {
    time_t tv_sec;
    int tv_usec; // нано
};
clock_gettime()
clock_gettime()
clock_gettime()
```

```
struct tm {}
time()
sleep()
```

Разрешение часов:
Вычислители семейства
Багет – 4 мкс
x86 –
rdtsc – время в тактах
процессора

Время в ОС RV

- Часы реального времени;
- создание однократных и периодических таймеров
- время первого срабатывания
 - От момента запуска
 - По значению часов
- интервал срабатывания

Принципы построения ОС РВ

- С монолитным ядром (как традиционный Unix)
 - проблемы с детерминированностью обработки прерываний и работы с драйверами устройств
- С микроядром
 - службы ОС оформляются как процессы

Примеры ОС РВ

Коммерческие

- ОС2000 (НИИСИ РАН)
<https://www.niisi.ru/otd07.htm>
- vxWorks (WindRiver Software, www.wrs.com)
- QNX (QNX Software)
- LynxOS (LunixWorks)
- Расширение Windows NT
- WinCE

Бесплатные с открытым исходным кодом

- RTEMS (www.rtems.org)
- uCOS

Расширения Linux

- RTLinux
- RTAI

Linux для ВС РВ (продолжение)

- RTlinux
- потоки РВ – в режиме ядра;
- обычное ядро Linux – задача низкого приоритета
- Требуется переработка драйверов для режима РВ

Литература

- И.Б. Бурдонов, А.С. Косачев, В.Н. Пономаренко. Операционные системы реального времени. – М.:ИСП РАН. – 2006. URL: http://www.ispras.ru/preprints/docs/prep_14_2006.pdf

Представительный обзор состояния дел в области ОС РВ.

- Bill O. Gallmeister. POSIX. 4: Programming for the Real World. – O'Reilly. – 1995. URL: <http://www2.fisica.unlp.edu.ar/electro/temas/p7/POSIX4.pdf>

Классическая книга по интерфейсам POSIX для систем РВ (ранее POSIX 1003.1 именовался POSIX.4)

- К.Е. Климентьев. Операционные системы реального времени. Обзорный курс лекций для студентов заочной формы обучения по специальности 230102 (Самарский государственный аэрокосмический университет). URL: <http://www.ssau.ru/files/resources/sotrudniki/rts2008.pdf>

Рассмотрен пример системы РВ и основные характеристики ОС РВ.

- C.L. Liu, J.W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment // JACM. – V.20. – No 1. – 1973. – PP. 46-61

Доказательство свойств алгоритмов планирования RMS и EDF

- Е.В. Ляпунцова. Системы реального времени. URL: <http://liapuntsova.ru/srv>

Ещё один курс лекций по ОС РВ

- URL: <http://www.rtsoft.ru/press/articles/detail.php?ID=1486>

Результаты измерения времени реакции ОС РВ на различных аппаратных платформах