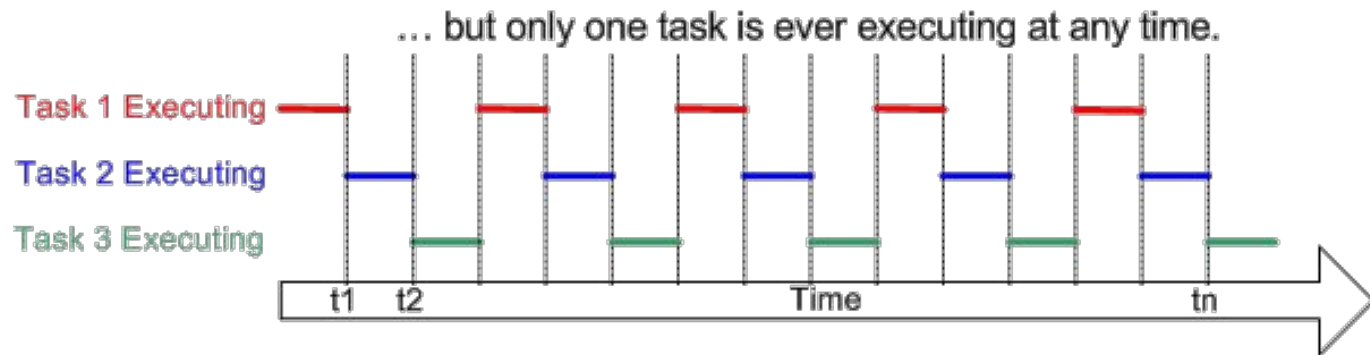
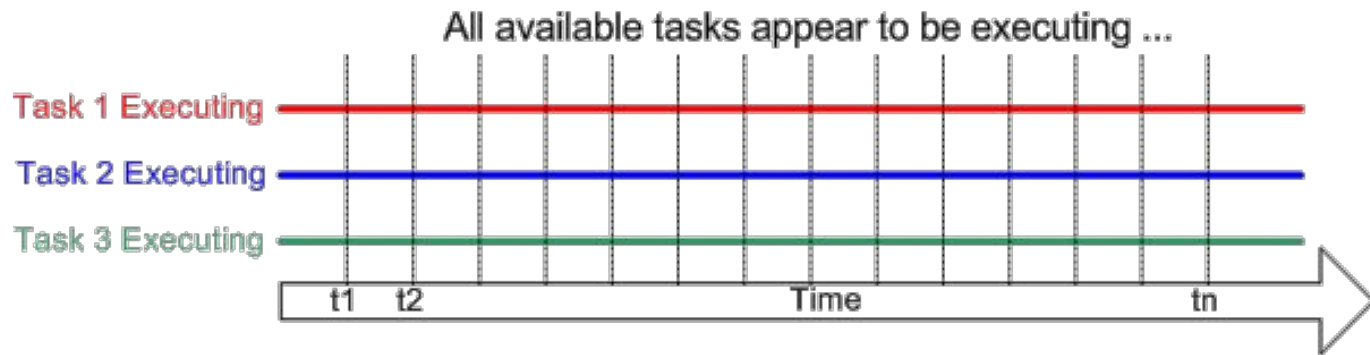


# Программирование микроконтроллеров STM32

*FreeRTOS and Tasks*

# Концепция ОС



# Минимальная ОС. Куда двигаться дальше?

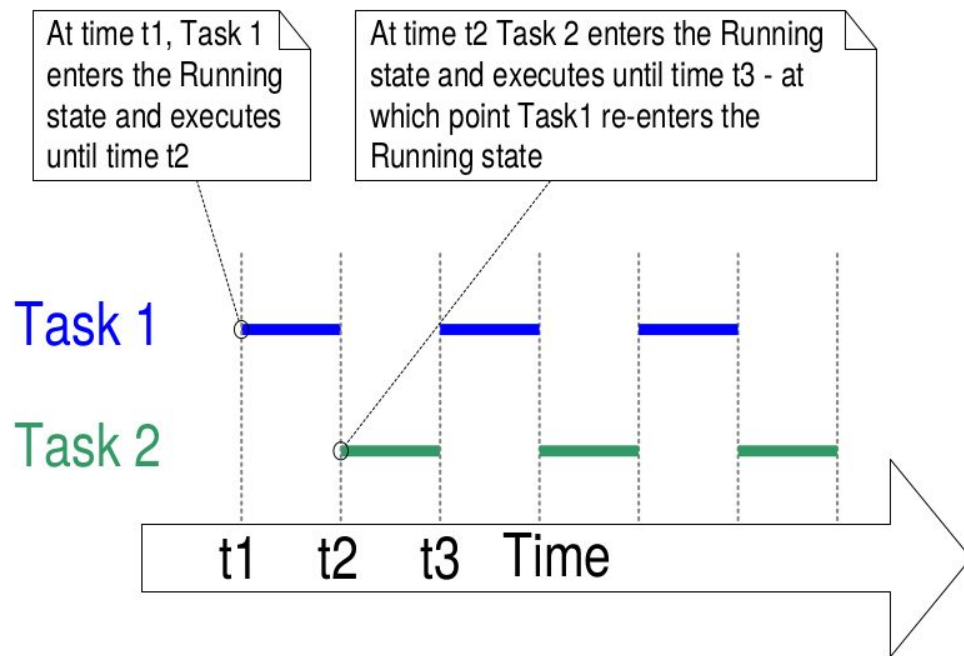


1. Добавление приоритетных задачам
2. Добавление менеджера памяти
3. Добавление средств межпроцессорного взаимодействия
4. Добавление поддержки системных вызовов для удобной работы с драйверами (не имеет смысла для Cortex-M0)
5. Анализ трассировки

# FreeRTOS. Планировщик

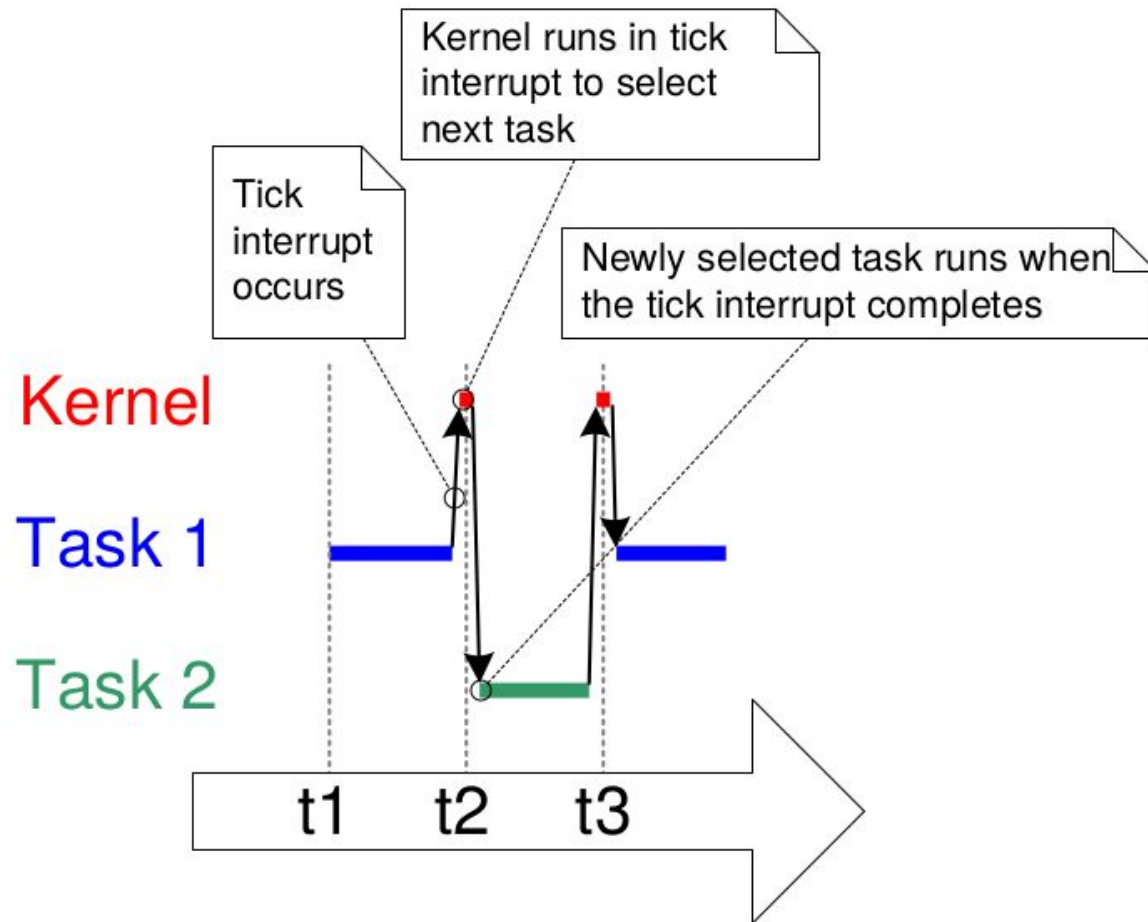
```
BaseType_t xTaskCreate(TaskFunction_t pvTaskCode,  
                        const char *const pcName,  
                        uint16_t usStackDepth,  
                        void *pvParameters,  
                        UBaseType_t uxPriority,  
                        TaskHandle_t *pxCreatedTask );  
  
void ATaskFunction(void *pvParameters);
```

# FreeRTOS. Планировщик



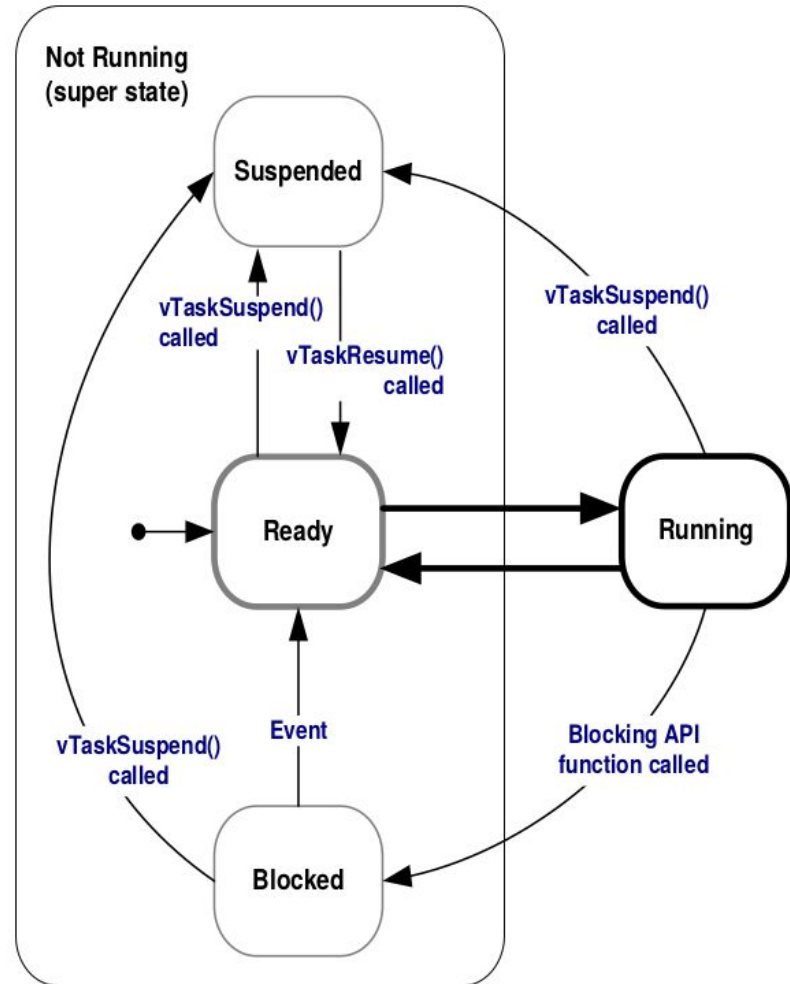
Задачи не могут  
выполняться параллельно  
на одном ядре  
микроконтроллера!

# FreeRTOS. Планировщик



# FreeRTOS. Планировщик

- Блокированное состояние (blocked state) - задержка или синхронизация
- Приостановленное состояние (suspended state) - задача недоступна планировщику



# FreeRTOS. Приоритеты

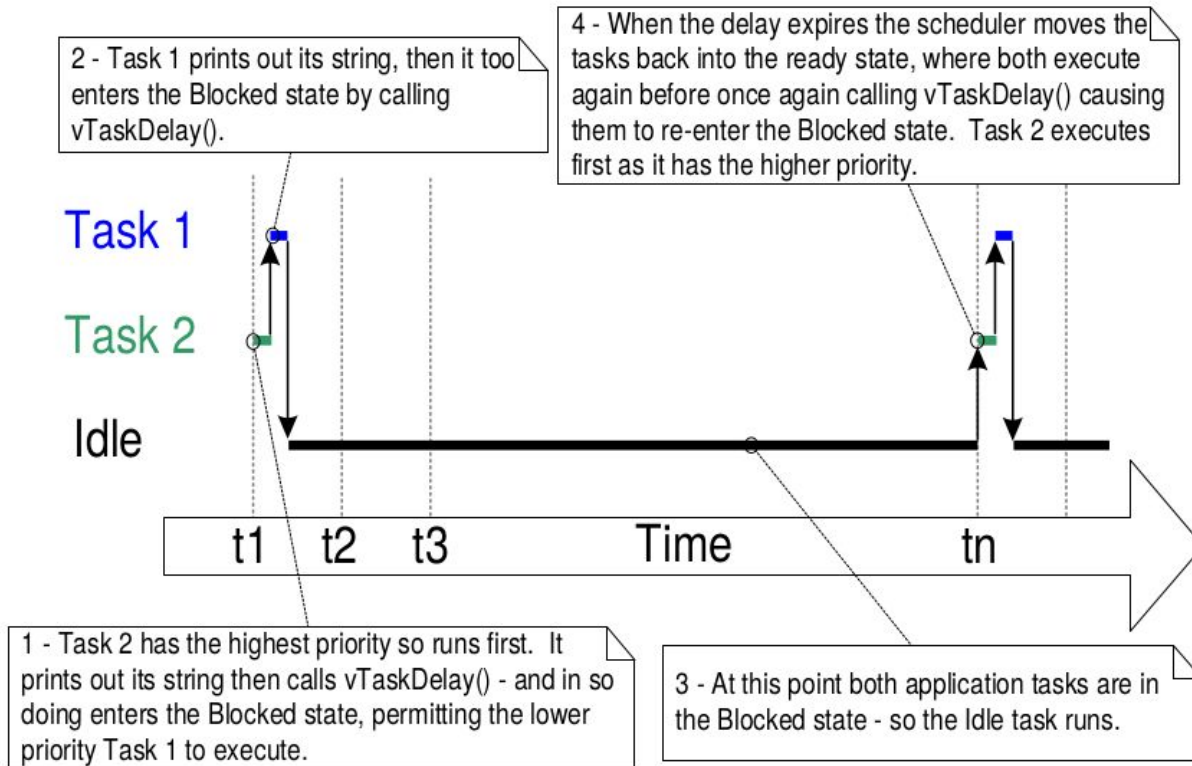
- От 0 до (`configMAX_PRIORITIES` – 1)
- Выше численный приоритет - выше логический приоритет
- Выше приоритет - больше времени на исполнение



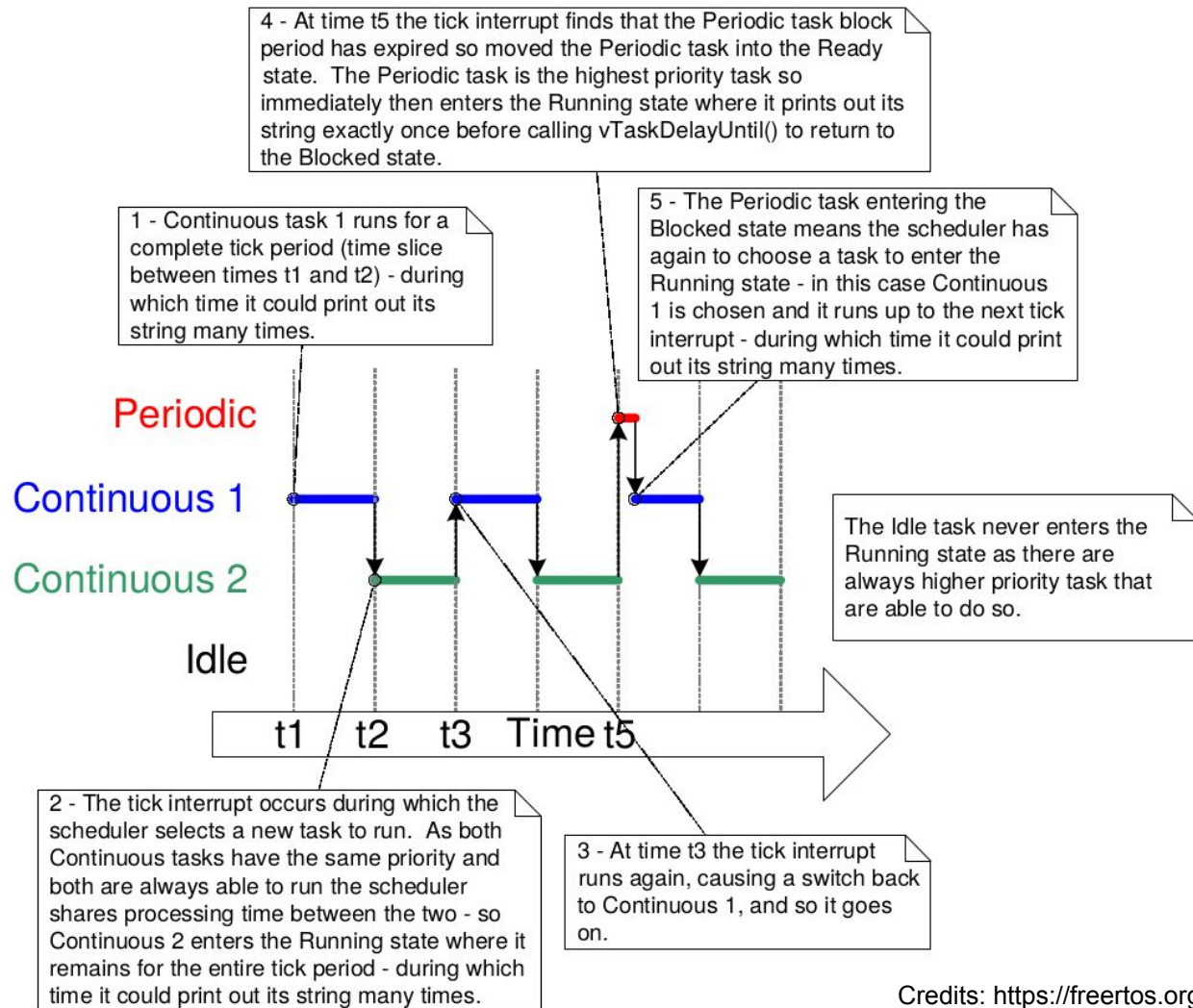
# FreeRTOS. Приоритеты

```
void vTaskFunction(void *pvParameters)
{
    char *pcTaskName;
    const TickType_t xDelay250ms = pdMS_TO_TICKS(250);
    for (;;) {
        vPrintString(pcTaskName);
        vTaskDelay(xDelay250ms);
    }
}
```

# FreeRTOS. Приоритеты



# FreeRTOS. Приоритеты



# FreeRTOS. Пустая задача (idle task)

- Бесконечный цикл, всегда в состоянии Ready
- Самый низкий приоритет
- Можно использовать для выполнения фоновый задач
- Не должна находиться в состоянии Blocked или Suspended (иначе не будет ни одной задачи на выполнение)
- `void vApplicationIdleHook(void);`

# FreeRTOS. Алгоритмы планировщика

configUSE_PREEMPTION	configUSE_TIME_SLICING	Алгоритм
1	1	Prioritized Pre-emptive Scheduling with Time Slicing
1	0	Prioritized Pre-emptive Scheduling (without Time Slicing)
0	Any	Co-operative Scheduling

# Репозиторий

[https://github.com/edosedgar/stm32f0\\_ARM](https://github.com/edosedgar/stm32f0_ARM)