

## EcoLab4

Реализация алгоритма планирования кратчайшая задача первая  
«Shortest Job First» (SJF) на эмуляторе QEMU.

1. Алгоритм
2. Реализация в QEMU

Выполнила  
Слепнева Мария  
21ПИ-3

## Алгоритм

Принцип работы алгоритма «Shortest Job First» (SJF) заключается в том, что процессы выбираются для выполнения исходя из их длительности. Процесс с самым коротким временем выполнения выбирается для выполнения первым, а остальные процессы ждут своей очереди. Этот метод позволяет максимизировать использование процессора за счет выполнения коротких процессов в первую очередь.

Моя реализация заключается в проходе по списку задач и выбора с наименьшей длительностью исполнения.

```
while (1) {
    size_t i = 0;
    min_duration = 0;
    for (i = 0; i < MAX_STATIC_TASK_COUNT; ++i) {
        if (pCMe->m_pTaskList[i].pfunc != 0 &&
            (pCMe->m_pTaskList[i].duration < min_duration || min_duration == 0)) {
            g_indx = i;
            min_duration = pCMe->m_pTaskList[i].duration;
        }
    }
    pCMe->m_pTaskList[g_indx].pfunc(min_duration);
    pCMe->m_pTaskList[g_indx].pfunc = 0;
    min_duration = 0;
    g_indx = 0;
}
```

## Реализация в QEMU

Для реализации использовался интерфейс Eco.Framework.

Было создано дополнительное поле в SEcoTask1Lab.h и добавлен аргумент для создания задачи

```
/* Данные экземпляра */
void (*pfunc) (uint64_t);

/* Длительность выполнения */
uint64_t duration;
```

При создании задачи в NewTask теперь передается поле duration для дальнейшей инициализации

```
int16_t (ECOCALLMETHOD *NewTask)(/*in*/ IEcoTaskScheduler1Ptr_t me, /*in*/ voidptr_t address, /*in*/ uint64_t duration,
/*in*/ uint32_t stackSize, /* out */ IEcoTask1** ppITask);
```

```

int16_t ECOCALLMETHOD CEcoTaskSchedulerLab_C761620F_NewTask(/*in*/ IEcoTaskSchedulerIPtr_t me, /*in*/ voidptr_t address, /*in*/ uint64_t duration,
/*CecTaskSchedulerLab_C761620F*/ pCHe = (CEcoTaskSchedulerLab_C761620F*)me;*/
int32_t indx = 0;
int32_t reg = 30;
uint64_t* pxTopOfStack = 0;

/* Проверка указателей */
if (me == 0 ) {
    return -1;
}

/* Проверяем указатель пула статических задач */
for (indx = 0; indx < 5; indx++) {
    if (g_xCEcoTaskList_C761620F[indx].pfunc == 0) {
        g_xCEcoTaskList_C761620F[indx].pfunc = address;
        g_xCEcoTaskList_C761620F[indx].duration = duration;
        g_xCEcoTaskList_C761620F[indx].m_cRet = 1;
        g_xCEcoTaskList_C761620F[indx].m_sp = (byte_t*)&g_xCEcoStackTaskList_C761620F[indx*4096];
        pxTopOfStack = g_xCEcoTaskList_C761620F[indx].m_sp;
    }
}

```

С целью более яркой визуализации были добавлены функции:

- PrintDuration
- PrintPercent
- printProgress

```

QEMU
Machine View
In process: Task 4
[*****] 92%Task duration: 4000000ms
EcoOS!

Task 1: Received!
Task 3: Received!
Task 4: Received!
Task 2: Received!
Task 8: Received!
Task 7: Received!
Task 6: Received!
Task 5: Received!

Task 3 on process => Task 3 completed
Task 5 on process => Task 5 completed
Task 8 on process => Task 8 completed
Task 4 on process

```

Демонстрацию работы можно посмотреть в файле demo.mp4