

EcoLab4

Реализация алгоритма планирования длинная задача первая
«Longest Job First» (LJF) на эмуляторе QEMU.

1. Алгоритм
2. Реализация в QEMU

Выполнил
Волков Владислав
21ПИ-3

Алгоритм

Принцип работы алгоритма "longest job first" (LJF) заключается в том, что процессы выбираются для выполнения исходя из их длительности. Процесс с самым длинным временем выполнения выбирается для выполнения первым, а остальные процессы ждут своей очереди. Этот метод позволяет максимизировать использование процессора за счет выполнения длинных процессов в первую очередь.

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

```
while (1) {
    size_t i = 0;
    uint64_t max_duration = 0;
    for (i = 0; i < MAX_STATIC_TASK_COUNT; ++i) {
        if (pCMe->m_pTaskList[i].pfunc != 0 && pCMe->m_pTaskList[i].duration > max_duration) {
            g_indx = i;
            max_duration = pCMe->m_pTaskList[i].duration;
        }
    }
    pCMe->m_pTaskList[g_indx].pfunc(max_duration);
    pCMe->m_pTaskList[g_indx].pfunc = 0;
    max_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);
```

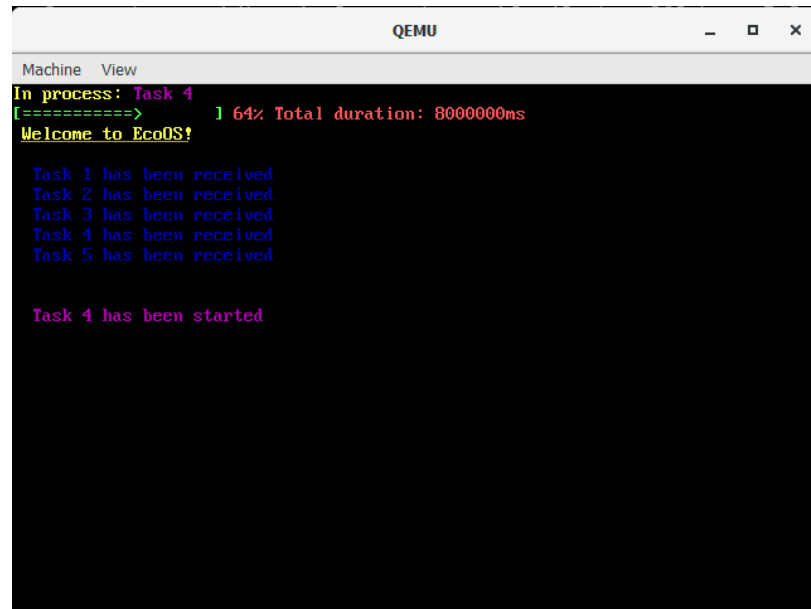
```
int16_t ECOCALLMETHOD CEcoTaskScheduler1Lab_C761620F_NewTask(/*in*/ IEcoTaskScheduler1Ptr_t me, /*in*/ voidptr_t address, /*in*/ uint64_t duration,
/*in*/ uint32_t stackSize, /* out */ IEcoTask1** ppITask) {
    /*CEcoTaskScheduler1Lab_C761620F* pCMe = (CEcoTaskScheduler1Lab_C761620F*)me;*/
    int32_t 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_CRef = 1;
            g_xCEcoTaskList_C761620F[indx].m_sp = (byte_t*)&g_xCEcoStackTaskList_C761620F[indx*4096];
            pxTopOfStack = g_xCEcoTaskList_C761620F[indx].m_sp;
            while (reg-- > 0) {
                g_xCEcoTaskList_C761620F[indx].m_sp = (byte_t*)&g_xCEcoStackTaskList_C761620F[indx*4096];
            }
        }
    }
    return 0;
}
```

Для более приятного вывода были реализованы функции:

- PrintDuration
- PrintPercent
- printProgress



```
QEMU
Machine View
In process: Task 4
[=====>] 64% Total duration: 8000000ms
Welcome to EcoOS!

Task 1 has been received
Task 2 has been received
Task 3 has been received
Task 4 has been received
Task 5 has been received

Task 4 has been started
```

Демонстрацию работы можно посмотреть в файле demo.mp4