



Mikrokontrolery ARM

Laboratorium 1

Pierczyk Krzysztof
2 kwietnia 2020



W ramach pracy laboratoryjnej napisany został krótki program sortujący tablicę w oparciu o język C++ oraz system operacyjny ISIX w wersji 3. Wykorzystany do tego został program demonstracyjny, przedstawiony na zajęciach, który został zmodyfikowany w następujący sposób:

- funkcja realizująca wątek została zastąpiona przez funkcję `void sort_thread()`
- z kodu funkcji `main()` zostały usunięte elementy odpowiedzialne za konfigurację portu sterującego diodą LED
- dopisana została funkcja `void selectionSort(int tab[], int size)` realizująca algorytm selection sort

Po utworzeniu wątku program zapisuje aktualny czas, który upłynął od startu programu, wywołuje funkcję sortującą dla tablicy `array_for_sort`, po czym ponownie mierzy czas, aby wypisać na port szeregowy czas sortowania. Kod realizujący to zadanie został przedstawiony poniżej:

```
namespace {
    void selectionSort(int tab[], int size){
        int i, j, k;
        for(i = 0; i < size; ++i) {
            k = i;
            for(j = i + 1; j < size; ++j)
                if(tab[j]<tab[k])
                    k = j;
            int temp = tab[k];
            tab[k] = tab[i];
            tab[i] = temp;
        }
    }
}

void sort_thread(void*){
    unsigned int size = 4096;
    osutick_t start = isix::get_ujiffies();
    selectionSort(array_for_sort, size);
    dbprintf("%i", isix::get_ujiffies() - start);
}
```

Analogiczny program został napisany dla hosta z wykorzystaniem lokalnego kompilatora C++. Jego kod jest identyczny za wyjątkiem funkcji realizujących pomiar czasu (tu posłużono się biblioteką `chrono`) oraz wypisywanie. Program został następnie uruchomiony 20 razy na każdej z platform, a czasy jego wykonywania zostały odnotowane.

	Host	Target
Jednostka obliczeniowa	Intel Core i5-7400 (3.00GHz)	STM32F411VET6 (Cortex-M4)
System operacyjny	Ubuntu 18.04.4 LTS	ISIX v.3
Kompilator	GCC 7.5.0	GCC 9.2.1

Tabela 1. Porównanie platform testowych

Ze względu na brak płytki ewaluacyjnej wyniki testów zostały zapożyczone od jednego z kolegów uczęszczających na laboratorium. Wyniki testów dla hosta zostały przedstawione w poniższej tabeli.

Numer testu	Wynik [μ s]
1	21768
2	23609
3	51088
4	21843
5	49717
6	22635
7	23197
8	23658
9	51121
10	49082
11	50668
12	22384
13	21762
14	51230
15	44701
16	21097
17	52751
18	61707
19	39647
20	63101

Tabela 2. Wyniki testów przeprowadzonych dla hosta

Wyniki te wykorzystano do policzenia kolejno średniego czasu wykonania i odchylenia standardowego tych czasów. Porównanie między obiema platformami zestawiono w Tabeli nr 3.

	Host	Target
Średni czas wykonania [μ s]	38338,3	1518118
Odchylenie standardowe [μ s]	15525,6	0

Tabela 3. Porównanie wyników obu platform

Zgodnie z oczekiwaniami średni czas wykonania zadania sortowania jest dużo (prawie 40-krotnie) mniejszy w przypadku PC. Wydajność jaką oferuje 64-bitowa jednostka Intelu jest nieporównywalnie wyższa niż ta prezentowana przez rdzeń Cortex-M4. Co jednak ciekawsze, chociaż też spodziewane, wyniki testów dla płytki ewaluacyjnej **były identyczne w przypadku wszystkich 20 prób**, kiedy na komputerze osobistym odchylenie standardowe miało wartość będącą niemalże połową wartości średniej. Jest to oczywiście wynik działania systemu operacyjnego. Niedeterministyczne przerwania od różnych elementów systemu wstrzymują wykonywanie programu na czas, którego nie da się a priori przewidzieć. W przypadku mikrokontrolera nawet, gdy działa on pod obsługą dedykowanego systemu operacyjnego, użytkownik ma niemal pełną kontrolę nad czasem wykonywania się poszczególnych fragmentów kodu. Zaprezentowane wyniki jednoznacznie pokazują przewagę Cortexa w przypadku **zastosowań, które wymagają dotrzymania pewnych ograniczeń czasowych**.