

Etap 1

- Opis programu

Program składa się z trzech etapów:

1. Tworzenie wskazanej ilości wątków i przypisanie im funkcjonalności

```
for(int i = 0; i<threads;i++)
{
    printf("[MAIN]: Tworzenie watku %d.\n", i);
    thread_tab[i] = i;
    pthread_create(&main_tab[i],NULL,function, &thread_tab[i]);
}
```

2. Watki wykonują nadaną im czynność i zostają uśpione

```
int index = ((int *) arguments)[0];
switch(index%func_amount)
{
    case 0:
        printf("[THREAD] %d: Watek rozpoczyna sortowanie losowych wartosci\n", index);
        sorting();
        break;
    case 1:
        printf("[THREAD] %d: Watek rozpoczyna sortowanie losowych wartosci\n", index);
        sorting_but_worse();
        break;
    case 2:
        printf("[THREAD] %d: Watek przenosi o losowa wartosc \n(wedlug algorytmu)\n", index);
        caesar_salad(index);
        break;
    case 3:
        printf("[THREAD] %d: Watek losuje dwie liczby w zakresie 0 - 100\n", index);
        rando(index);
        break;
    default:
        printf("[THREAD] %d: Watek nie dostal funkcji. Watek zostanie uspiony\n", index);
        break;
}

shleep(index);
```

3. Uśpione wątki zostają złączone z głównym wątkiem (zakończenie istnienia wątku)

```
for (int i = 0; i < threads; i++)
{
    pthread_join(main_tab[i], NULL);
    printf("[MAIN]: Watek %d zakonczony.\n", i);
}
free(main_tab);
free(thread_tab);
```

Na drugim zrzucie ekranu koda można zauważyć 4 możliwe funkcje, jakie mogą zostać nadane wątkowi.

Sorting() – sortowanie rosnąco losowo wygenerowanych liczb za pomocą algorytmu bąbelkowego

Sorting_but_worse() – sortowanie malejąco losowo wygenerowanych liczb za pomocą algorytmu bąbelkowego. Liczby znajdujące się w sortowanej tablicy są ustawione rosnąco, przez co przewidywana złożoność czasowa algorytmu wynosi $O(n^2)$

Caesar_salad() – wątek generuje losowy ciąg znaków składający się z 10 wielkich liter, następnie dokonuje przesunięcia o losową wartość. Wartość przesunięcia jest stała dla wszystkich pozycji w ciągu.

Rando() – wątek generuje liczbę z zakresu 0 – 100000000, którą następnie musi „odgadnąć” w sposób losowy. Po odgadnięciu liczby, zostaje ona wypisana.

Po wykonaniu przydanej funkcji, wątek zostaje uśpiony na czas w zakresie 0 – 10 sekund (wartość jest nadawana losowo). Po złączeniu wszystkich wątków, pamięć zarezerwowana dynamicznie zostaje zwolniona.

- Sposób kompilacji

```
/Documents/PWR/sys-opr-2p/etap_1$ gcc -pthread main.c -o main
```

Program jest kompilowany za pomocą programu gcc. Program jest zapisywany pod nazwą main. -pthread służy do „zlinkowania” biblioteki pthreads.

- Przykładowy wynik działania programu.

```
[MAIN] Podaj ilość porzadzanych watkow: 3
[MAIN]: Tworzenie watku 0.
[MAIN]: Tworzenie watku 1.
[MAIN]: Tworzenie watku 2.
[MAIN]: Wszystkie watki zostaly stworzone.
[THREAD] 1: Watek rozpoczyna sortowanie losowych wartosci o nieoptymalnym rozlozeniu
[THREAD] 1:[0]: 138[1]: 106[2]: 99[3]: 98[4]: 97[5]: 89[6]: 78[7]: 52[8]: 50[9]: 49
[THREAD] 1: Usypianie na 9 sekundy.
[THREAD] 0: Watek rozpoczyna sortowanie losowych wartosci
[THREAD] 0:[0]: 23[1]: 40[2]: 41[3]: 41[4]: 49[5]: 54[6]: 69[7]: 80[8]: 88[9]: 100
[THREAD] 0: Usypianie na 2 sekundy.
[THREAD] 2: Watek przenosi o losowa wartosc (wedlug tablicy ascii) wszystkie znaki z podanego przez uzytkownika slowa:
[THREAD] 2: poczatkowe slowo: HPBKZLF[WC
[THREAD] 2: Slowo zostalo zmienione na: LTF0^PJ_G
[THREAD] 2: Usypianie na 10 sekundy.

[THREAD] 0: Zakonczone.
[MAIN]: Watek 0 zakonczony.

[THREAD] 1: Zakonczone.
[MAIN]: Watek 1 zakonczony.

[THREAD] 2: Zakonczone.
[MAIN]: Watek 2 zakonczony.
ENDlightiy@ubuntu:~/Documents/PWR/sys-opr-2p/etap_1$
```

Czynności, które są wykonywane w sekcji main programu zaczynają się od etykiety „[MAIN]”, natomiast czynności wątków są opisywane przez komunikaty z etykietą „[THREAD] x:”, gdzie x odpowiada id wątku. Zakończenie programu symbolizuje komunikat „END”.