

Zadanie 1

Po skompilowaniu i uruchomieniu programu okazało się, że wątki wykonują się w losowej kolejności.

Zadanie 2

Po poprawieniu kodu w pliku *hello32.c* (dodanie rzutowania zmiennej na typ **long**) i jego kompilacji poleceniem *gcc hello32.c -lm -pthread*, uruchomiono uzyskany program trzy razy dla porównania jego wyników.

```
main(): Created 32 threads.
2: Hello World!
5: Hello World!
11: Hello World!
12: Hello World!
17: Hello World!
13: Hello World!
19: Hello World!
21: Hello World!
23: Hello World!
24: Hello World!
0: Hello World!
4: Hello World!
26: Hello World!
27: Hello World!
28: Hello World!
30: Hello World!
31: Hello World!
14: Hello World!
16: Hello World!
18: Hello World!
20: Hello World!
29: Hello World!
1: Hello World!
9: Hello World!
6: Hello World!
15: Hello World!
10: Hello World!
22: Hello World!
25: Hello World!
7: Hello World!
3: Hello World!
8: Hello World!
```

```
main(): Created 32 threads.
12: Hello World!
29: Hello World!
20: Hello World!
25: Hello World!
31: Hello World!
8: Hello World!
28: Hello World!
21: Hello World!
7: Hello World!
22: Hello World!
18: Hello World!
4: Hello World!
23: Hello World!
14: Hello World!
6: Hello World!
9: Hello World!
11: Hello World!
5: Hello World!
3: Hello World!
2: Hello World!
13: Hello World!
1: Hello World!
17: Hello World!
24: Hello World!
16: Hello World!
0: Hello World!
27: Hello World!
30: Hello World!
15: Hello World!
19: Hello World!
26: Hello World!
10: Hello World!
```

```
main(): Created 32 threads.
10: Hello World!
1: Hello World!
17: Hello World!
14: Hello World!
13: Hello World!
19: Hello World!
21: Hello World!
24: Hello World!
3: Hello World!
25: Hello World!
26: Hello World!
28: Hello World!
29: Hello World!
31: Hello World!
0: Hello World!
23: Hello World!
15: Hello World!
12: Hello World!
27: Hello World!
4: Hello World!
20: Hello World!
18: Hello World!
22: Hello World!
11: Hello World!
30: Hello World!
16: Hello World!
5: Hello World!
8: Hello World!
2: Hello World!
6: Hello World!
7: Hello World!
9: Hello World!
```

Wyniki uruchomienia programu.

Kolejność wykonywania wątków jest praktycznie losowa i wynika z algorytmu szeregowania zadań systemu.

Zadanie 3

W programie *hello_arg1.c* użyto tablicy do uzależnienia tego, co jest wypisywane na *stdout* od ID wątku. Przy tworzeniu wątku przekazywany jest argument w postaci elementu tablicy będącym ID wątku.

W programie *hello_arg2.c* użyto struktury *thread_data* do przekazania wielu argumentów wątkowi. Przy tworzeniu wątku przekazywany jest argument w postaci elementu tablicy, będącym strukturą odpowiadającą wątkowi.

Program *bug3.c* zadziałał poprawnie po zmianie 19 i 32 linijki. W tej pierwszej do zmiennej *taskid* przypisywany był wskaźnik na wskaźnik na zmienną *long*. Zmieniono rzutowanie i przypisanie w taki sposób, aby typy zmiennych się zgadzały oraz aby do *taskid* poprawnie przypisane było ID wątku. Kolejny błąd pojawił

się przy tworzeniu wątku, jako argument przekazywany był adres zmiennej *t*. Program działał poprawnie po usunięciu znaku *&*.

Zadanie 4

Program kończy swoje działanie, zanim wątki wykonają się do końca. Jest to spowodowane tym, że obliczenia, które muszą wykonać wątki są czasochłonne i brakiem funkcji *pthread_exit()* lub *pthread_join()*, które zapobiegają przedczesnemu zakończeniu pracy programu. Program działa poprawnie po dodaniu funkcji *pthread_exit()* na samym końcu programu głównego