
PROGRAMOWANIE WĄTKÓW - PODSTAWOWE MECHANIZMY

JAKUB SZCZYRK
235477

PONIEDZIAŁEK 13:15-14:00

20 STYCZNIA 2019

Spis treści

1	Zadanie 1 hello.c.	2
2	Zadanie 2 hello32.c.	2
3	Zadanie 3	2
3.1	hello_arg1.c	2
3.2	hello_arg2.c	2
3.3	bug3.c	3
4	Zadanie 4 bug5.c	3
5	Zadanie 5	3
5.1	join.c.	3
5.2	detached.c.	4
6	Zadanie 6 bug2.c	4

1 Zadanie 1 hello.c.

Program tworzy pięć wątków za pomocą komendy `pthread_create`, po utworzeniu wyświetla powitalny komunikat. Podczas działania programu można zauważyć, że wątki tworzą się w odpowiedniej kolejności lecz ich komunikat nie jest natychmiastowy.

```
int pthread_create(pthread_t * thread, pthread_attr_t *attr,
                  void * (*start_routine)(void *) fun, void * arg);
```

Funkcja `pthread_create` przyjmuje następujące argumenty:

- `thread` - identyfikator wątku struktura typu `pthread_t` zostaje zwrócona przez funkcję jako identyfikator utworzonego wątku.
- `attr` - wskaźnik na atrybuty wątku, określa szczegóły dotyczące wątku; `NULL` ustawia domyślne wartości;
- `fun` - funkcja wykonywana w wątku; przyjmuje argument typu `void*` i zwraca wartość tego samego typu; `void*` może przekonwertować na dowolny inny typ wskaźnika bez wyraźnego rzutowania. Nie można na nim wykonywać działań arytmetycznych;
- `arg` - argument przekazywany do funkcji. typ `void*` jest użyty w tej funkcji ze względu na to, że może pracować z różnymi typami wskaźników.

Funkcja `pthread_exit` kończy działanie wątku, który wywołał tę funkcję. Zakończenie w taki sposób, nawet głównego wątku nie wpłynie w żaden sposób na inne wątki.

2 Zadanie 2 hello32.c.

Program tworzy trzydzieści dwa wątki za pomocą komendy `pthread_create`, po utworzeniu wyświetla powitalny komunikat. Podczas działania programu można zauważyć, że wątki w losowej kolejności wyświetlają komunikat, jest to spowodowane tym, że każdy z nich wykonuje funkcję `sleep` w celu współbieżnego działania, a następnie wykonują czasochłonne operacje matematyczne co powoduje zakończenie różnych wątków. Program jest przykładem zachowania się planisty systemu operacyjnego na kolejności wykonywania wątków.

3 Zadanie 3

3.1 hello_arg1.c

Program tworzy osiem wątków za pomocą komendy `pthread_create`, po utworzeniu wyświetla powitalny komunikat dla danego kraju. Podczas działania programu można zauważyć, że wątki są tworzone w odpowiedniej kolejności, ale w losowej wyświetlają komunikat, mamy tu również wykorzystaną funkcję `sleep` w celu współbieżnego działania. Program demonstruje jeden z bezpiecznych sposobów przekazywania pojedynczego argumentu do wątków podczas tworzenia wątku, wykorzystując wcześniej przygotowaną tablicę.

3.2 hello_arg2.c

Program tworzy osiem wątków za pomocą komendy `pthread_create`, po utworzeniu wyświetla powitalny komunikat dla danego kraju. Podczas działania programu można zauważyć, że

wątki są tworzone w odpowiedniej kolejności, ale w losowej wyświetlają komunikat , mamy tu również wykorzystaną funkcję `sleep` w celu współbieżnego działania. Program demonstruje jeden z bezpiecznych sposobów przekazywania argumentów do wątków podczas tworzenia wątku, przekazując strukturę jako zbiór danych.

3.3 bug3.c

Program tworzy osiem wątków za pomocą komendy `pthread_create`, po utworzeniu wyświetla komunikat. Podczas działania programu można zauważyć, że wątki są tworzone w odpowiedniej kolejności, ale wyświetlają taki sam komunikat , mamy tu również wykorzystaną funkcję `sleep` w celu współbieżnego działania. Program demonstruje jeden z niebezpiecznych sposobów przekazywania argumentów do wątków podczas tworzenia wątku, poprzez przekazanie adresu do zmiennej ("t"), na której jest zapisana ostatnia zmiana wartości.

4 Zadanie 4 bug5.c

Program tworzy pięć wątków za pomocą komendy `pthread_create`, po utworzeniu wyświetla komunikat startu pracy złożonych działań arytmetycznych, a o skończonym zadaniu informuje. Podczas działania programu można zauważyć, że wątki są tworzone i wyświetlają komunikat w odpowiedniej kolejności. Program demonstruje zakończenie głównego wątku przed ostatnim utworzonym wątkiem, jest to spowodowane zbyt dużym nakładem obliczeniowym. Aby zapobiec temu problemowi należy na końcu głównego programu dodać funkcję `pthread_exit`.

5 Zadanie 5

5.1 join.c.

Program tworzy cztery wątki za pomocą komendy `pthread_create`, po utworzeniu wyświetla komunikat startu pracy złożonych działań arytmetycznych, a o skończonym zadaniu informuje. Wątek główny zapewnia procedurę łączenia za pomocą komendy `pthread_join` oraz informuje o zamoczeniu tego działania i działania całego programu. Podczas działania programu można zauważyć, że wątki są tworzone w odpowiedniej kolejności i wyświetlają komunikat. Program demonstruje tak zwane "czekanie" na uzupełnienie wątku.

Opis funkcji:

- `pthread_attr_init`
Funkcja inicjuje zmienna przed użyciem atrybutu.
- `pthread_attr_setdetachstate`
Funkcja `pthread_attr_setdetachstate` ustawia rodzaj wątku, jaki ma zostać ustalony przy jego tworzeniu. Rodzaj jest identyfikowany jedną z wartości:
 - `PTHREAD_CREATE_JOINABLE` - utworzenie wątku typu joinable (domyślnie);
 - `PTHREAD_CREATE_DETACHED` - utworzenie wątku typu detached.
- `pthread_attr_destroy`
Funkcja realizuje zwolnienie zasobów związanych z atrybutem.
- `pthread_join`
Funkcja implementuje oczekiwanie na zakończenie wątku jest dodatkowym sposobem

synchronizacji między wątkami i dotyczy wyłącznie wątków typu joinable. Wątek wywołujący `pthread_join` zostaje wstrzymany do chwili, gdy wskazany wątek zakończy działanie - wówczas wątek oczekujący jest kontynuowany, a `pthread_join` przekazuje wartość wynikową zwróconą przez wątek. Jeśli wątek został przerwany wartością zwracaną jest stała `PTHREAD_CANCELED`. Opis argumentów:

- `id` - identyfikator wątku, którego zakończenie jest oczekiwane;
- `retval` - wskaźnik na wartość wynikową wątku; może być `NULL`, wówczas wynik jest ignorowany

5.2 `detached.c`.

Program tworzy cztery wątków za pomocą komendy `pthread_create`, po utworzeniu wyświetla komunikat startu pracy złożonych działań arytmetycznych, a o skończonym zadaniu informuje. Podczas działania programu można zauważyć, że wątki są tworzone w odpowiedniej kolejności i wyświetlają komunikat. Program demonstruje jawne tworzenie wątku w stanie odłączonym. Można to zrobić, w celu zaoszczędzenia niektórych zasoby systemowych, jeśli wątek nigdy nie będzie musiał się później połączyć.

6 Zadanie 6 `bug2.c`

Program tworzy osiem wątków za pomocą komendy `pthread_create`, po utworzeniu zwraca błąd "Segmentation fault", z powodu wyczerpania przestrzeni stosu wątków, ze względu na pracę złożonych działań arytmetycznych. W celu poprawienia błędu należy zwiększyć domyślny rozmiar przestrzeni stosu wątków co zostało zaimplementowane w programie `bug2fix.c`. Program demonstruje jeden ze sposobów naprawy.