

# Systemy operacyjne 2

## Wytyczne projektowe

Jarosław Rudy

9 kwietnia 2017

### 1 Projekt

Projekt polega na napisaniu aplikacji wielowątkowej na wybrany przez studenta temat. Tematyka jest (prawie) dowolna o ile założenia “techniczne” projektu są spełnione tzn. założenia nie muszą być zawsze zgodne ze znaną nam rzeczywistością. Celem projektu jest nabycie podstawowych umiejętności pisania aplikacji wielowątkowych oraz unikania problemów związanych z programowaniem współbieżnym. Obejmuje to:

- tworzenie i uruchamianie wątków,
- poprawne kończenie wątków i zamykanie aplikacji (zamknięcie wszystkich wątków przed zamknięciem wątku `main()`),
- rozwiązywanie problemu dostępu do zasobów (problem konta bankowego), czyli odpowiednie zdefiniowanie sekcji krytycznych (użycie mutexów i ewentualnie zmiennych warunkowych),
- rozwiązywanie lub zapobieganie zagłodzeniom,
- rozwiązywanie lub zapobieganie zakleszczeniom (deadlock).

Tematy projektów nie mogą być trywialne tj. niedopuszczalne są tematy typu “problem ucztyjących filozofów” czy prosty przykład “producent-konsument”, chyba że tematy te zostaną odpowiednio rozszerzone. Wybór tematu następuje poprzez przesłanie opisu tematu (patrz następne 2 akapity) w formacie PDF na adres e-mail prowadzącego i uzyskanie zgody prowadzącego na realizację (mailowo lub ustnie). Tematy powinny być różne. Decyduje kolejność zgłoszeń.

Opis tematu nie powinien przekraczać 2 stron i powinien zawierać m.in. dane studenta oraz tytuł. Ponadto, w opisie należy określić rodzaje i (w przybliżeniu) liczbę wątków. Najlepiej zdefiniować przynajmniej 2 lub 3 rodzaje wątków (poprzez rodzaje rozumiemy wątki realizujące inne zadania, np. wątki klientów, kelnerów, kucharzy itp.). Liczba wątków jest zależna od tematyki: projekty w stylu “producent-konsument” zwykle nie przekraczają 10 wątków, podczas gdy tematy typu “restauracja” mogą mieć nawet 100 wątków (nawet przy jednym procesorze/rdzeniu w komputerze). Należy zwięźle opisać rolę (działanie) poszczególnych typów wątków.

Tak samo należy określić rodzaje, liczbę i zastosowanie zasobów. Należy pamiętać, że zasoby zwykły odzwierciedlają aspekty statyczne (stan, dane), podczas gdy wątki są dynamiczne (akcje, czynności). Zwykle jeden “obiekt” nie może być jednocześnie wątkiem i zasobem. Przykładowo “miasto” można traktować jako zasób (“stan magazynów”, “liczba ludności”) lub jako wątek (symulacja zachowania miasta) lub jako dwa osobne obiekty wątek + zasób.

Do realizacji projektu należy wykorzystać linuksową bibliotekę pthreads (przez którą używamy funkcji C/C++ do obsługi wątków, a nie systemowych mechanizmów typu semafor systemowy lub kolejki komunikatów, które służą do komunikacji międzyprocesowej). Można też wykorzystać narzędzia wielowątkowe wbudowane w standard C++11 (może być konieczne określenie dodatkowych flag kompilatora). Aplikacja ma być napisana w języku C lub C++.

Istotną sprawą jest wizualizacja pracy wątków/stanu zasobów. Z jednej strony zaopatrzenie aplikacji w zaawansowany interfejs graficzny może być trudne i czasochłonne. Z drugiej strony domyślna konsola (terminal) nie nadaje się do wizualizacji. Zalecanym rozwiązaniem pośrednim jest wykorzystanie biblioteki `ncurses`, która umożliwia dostęp do dowolnego miejsca terminala w trybie tekstowym wraz z obsługą kolorów. Biblioteka ma niewiele miejsc na kolory, ale definicje poszczególnych kolorów można zmieniać, by uzyskać szerszą paletę barw. Do uzyskania takich kolorów konieczne być może ustawienie przed uruchomieniem programu pewnych zmiennych systemowych (np. poprzez wywołanie `TERM=xterm-256color`).

Wybrana wizualizacja powinna umożliwiać śledzenie zasobów i ich stanu (np. wypełnienie magazynu, liczba dostępnych narzędzi, wolnych miejsc parkingowych itp.). Przy małej liczbie wątków/zasobów można też pokusić się o dodanie informacji, który wątek aktualnie zajmuje dany zasób. Wizualizacja powinna też umożliwiać śledzenie położenia lub pracy wątków. Wątki nie mogą pracować na “pełnej” szybkości, gdyż utrudnia to ich śledzenie, a czasami może wręcz doprowadzić do zajęcia procesora przez jeden wątek. Problem ten rozwiązuje się poprzez usypianie wątków na określony czas (zwykle kilka do kilkuset milisekund) pomiędzy poszczególnymi akcjami (przebiegami głównej pętli wątku). Nie należy usypiać wątków na zbyt długi czas (1 sekunda i dłużej). W szczególności usypienie wszystkich wątków “jednocześnie” na czas 1 sekundy spowoduje “skokowe” działanie aplikacji (wszystkie wątki zmieniają stan jednocześnie), co utrudnia jej analizę.

**Uściślenie dodane 9.04.2017:** wątki powinny wykonywać jakąś pracę poza samym zajmowaniem i zwalnianiem zasobów. Praca to powinna być widoczna na ekranie w miarę płynnie (zmiana stanu/pozycji wątku co najmniej 2 razy na sekundę). Do symulowania “płynnego przebiegu” niektórych czynności (np. ścinanie drzewa) można użyć wskaźników postępu: paska lub oznaczenia liczbowego (np. od 00 do 99), które są zmieniane płynnie. Jedynym wyjątkiem jest sytuacja, gdy wątek rzeczywiście nic nie robi (np. czekając na jakieś wydarzenie). Oznacza to, że wątek zablokowany na wywołaniu `wait` nie musi zmieniać swojego stanu na ekranie.