Problem palaczy tytoniu

Bartosz Bednarczyk

22 stycznia 2016

Wymagania techniczne

Rozwiązanie powinno być napisane w języku C i kompilować się bez błędów z opcjami **-std=gnu99 -Wall -Wextra** kompilatorem **gcc** lub **clang** w systemie **Linux**. Do rozwiązania musi być dostarczony plik **Makefile** tak, aby po wywołaniu polecenia **make** otrzymać pliki binarne, a polecenie **make clean** powinno zostawić w katalogu tylko pliki źródłowe.

Treść zadania

Treść zadania

Przypuśćmy, że istnieją trzy procesy palaczy i jeden proces agenta. Każdy z palaczy na okrągło robi papierosy i je pali. Zrobienie i zapalenie papierosa wymaga tytoniu, bibułki i zapałek. Każdy palacz posiada nieskończoną ilość wyłącznie jednego typu zasobu, tj. pierwszy ma tytoń, drugi bibułki, a trzeci zapałki. Agent kładzie na stole dwa składniki. Palacz, który ma brakujący składnik, podnosi ze stołu resztę, skręca papierosa i go zapala. Agent czeka, aż palacz skończy palić, po czym powtarza cykl.

Historia problemu

Problem palaczy tytoniu został po raz pierwszy przedstawiony przez Suhasa Patila w 1971 roku, który uważał, że nie da się go rozwiązać przy pomocy semaforów. Krótko opiszę trzy wersje problemu.

W pierwszej wersji Patil zakładał dodatkowe ograniczenie na rozwiązanie. Po pierwsze, nie wolno było modyfikować kodu agenta. Jeśli agent miałby reprezentować system operacyjny, to sensownie jest założyć, że nie chcemy go zmieniać z każdą nową aplikacją. Drugim ograniczeniem jest to, że nie możemy używać instrukcji warunkowych lub tablic semaforów. Przy takich ograniczeniach problem nie może być rozwiązany, ale jak Pernas wskazuje, drugie ograniczenie jest dość sztuczne. Z ograniczeniami jak te, wiele problemów synchronizacyjnych okazuje się być nierozwiązywalne.

W drugiej, ciekawszej wersji problemu, pomijamy drugie z powyższych założeń.

Trzecia wersja problemu polega na tym, że agent informuje palacza o możliwości zrobienia papierosa. Ten przypadek jest niepraktyczny, gdyż wymaga od agenta informacji na temat innych palaczy. Dodatkowo założenie o składnikach papierosów jest wtedy nieistotne. Ponadto, przedstawiona wersja jest zbyt łatwa, by nadawać się na pracownię z systemów operacyjnych.

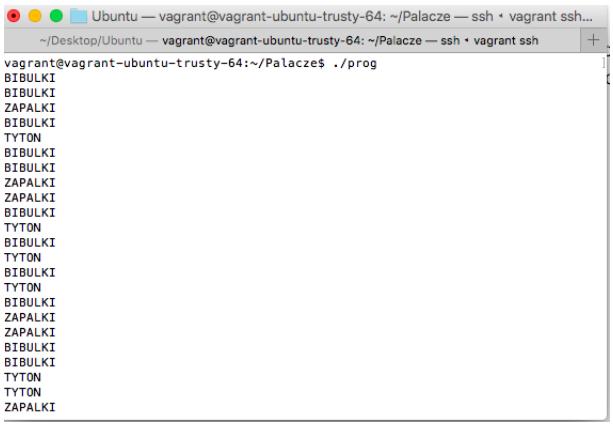
Potencjalne problemy

Podczas rozwiązywania zadania można napotkać się z następującymi problemami:

- zabranie składników ze stołu przez dwóch różnych palaczy;
- kelner nie czeka, aż palacz skończy palić papierosa.

Rozwiązanie

Rozwiązanie umieszczone jest w katalogu „Palacze" dołączonym do sprawozdania.
Opis algorytmu i szczegóły implementacji można łatwo wyczytać z kodu źródłowego.
Przykładowy wynik uruchomienia programu możemy zaobserwować na poniższym zdjęciu:



Słowa „BIBUŁKI", „TYTON", „ZAPALKI" oznaczają, że palacz, który posiada wypisany przez program zasób, zapalił papierosa. Program działa do czasu, aż nie zostanie sam wyłączony przez użytkownika.

Testy

Program uruchamiany był wielokrotnie i podczas jego uruchomień zaobserwowałem, że działa on płynnie i bez opóźnień. Sugeruje to, że nie dochodzi do zakleszczeń procesów.

Przeprowadziłem również badanie polegające na porównaniu liczby wystąpień odpowiednich wyrazów. Po zatrzymaniu programu, wypisał on 11052 wyrazy. Spośród nich 3660 to „TYTON", 3712 to „ZAPALKI", a 3680 to „BIBULKI". Sugeruje to, iż rozłożenie aktywności różnych palaczy w czasie jest w miarę równomierne.