

Systemy Operacyjne 2

Zadanie 1: Problem uczujących filozofów

Prowadzący: dr inż. Dominik Żelazny

Notatka z wykonania:

1. Wprowadzenie

Problem uczujących filozofów jest jednym z klasyków informatyki. Zapoczątkował go Edsger Dijkstra, aby zobrazować próby dostania się procesów systemów operacyjnych (lub komputerów) do zasobów. C.A.R. Hoare sparafrazował problem do wersji znanej dzisiaj. Tłumaczy on oraz obrazuje problem zakleszczenia jednostek. Dotyczy sytuacji, w której pięciu filozofów siedzi wokół okrągłego stołu i na zmianę jedzą i myślą. Rozwiązanie problemu to zmaksymalizowany czas jedzenia oraz myślenia każdego z uczonych.

2. Wersja problemu

Program jest implementacją problemu uczujących filozofów dla pięciu filozofów siedzących wokół okrągłego stołu i jedzących za pomocą sztućców, w programie opisanych jako pałeczki. Każdy filozof ma przed sobą talerz z jedzeniem oraz dwa sztucce po obu stronach naczynia, ale nie może zacząć jeść, dopóki nie zdobędzie obu (stąd zakleszczenie – filozof może podnieść jeden sztuciec, ale niekoniecznie drugi). Ponadto filozofowie nie mogą rozmawiać ze sobą i nie mają możliwości pomocy innym. Głównym celem problemu jest uniknięcie blokowania się uczonych nawzajem oraz zmaksymalizowanie czasu jedzenia oraz myślenia każdego z nich.

3. Użyty język programowania

Program został napisany w języku C++.

4. Rozwiązanie problemu

Program używa wątków, *mutexów* i biblioteki *ncurses* do implementacji problemu. Każdy filozof jest reprezentowany przez oddzielny wątek, który cyklicznie wykonuje następujące czynności: myśli przez pewien czas, próbuje podnieść dwie pałeczki (czyli zablokować dwa *mutexy* reprezentujące sztucce) i jeść, jedząc przez pewien czas, odkłada sztucce (czyli zwalnia *mutexy* reprezentujące sztucce) i zaczyna myśleć ponownie.

Wszystkie *mutexy* są reprezentowane przez tablicę pałeczek, a czas, przez jaki każdy filozof je i myśli, jest monitorowany przez tablice sekund indeksowane „numerem” filozofa – każdy z tych indeksów odpowiada jednemu imieniu znanego uczonego, aby zhumanizować *output*.

Program korzysta również z funkcji *ncurses* do wyświetlania stanu każdego filozofa na bieżąco na konsoli. Po minięciu czasu symulacji wyniki są wyświetlane w postaci *"Filozof X zjadł Y sekund i myślał przez Z sekund"*.

5. Podsumowanie

Program przedstawia rozwiązanie problemu uczujących filozofów z użyciem wątków, *mutexów* i biblioteki *ncurses*. Program może być używany do symulacji większej liczby filozofów, jeśli tylko zmienione zostaną wartości stałych w kodzie. Badania zostały jednak przeprowadzone jednak na stałych wartościach, tak jak w klasycznym opisie problemu.