Paradygmaty programowania - ćwiczenia
Lista 12

Wszystkie programy mają być napisane w języku Java.

1. Problem ucztujących filozofów

Problem polega na zsynchronizowaniu działań pięciu chińskich filozofów, którzy mieszkają w położonej na odludziu chacie (według innych przekazów była to wieża z kości słoniowej), mieszczącej jadalnię i salę do medytacji. Żyją oni, starając się zachować równowagę między yang i yin, Niebem i Ziemią, duchem i ciałem. Filozofowie są na tyle zaawansowani w ascezie, że nie muszą spać ani wykonywać innych przyziemnych czynności (rzecz dzieje się u kresu dziesięciu mitycznych epok, prawdopodobnie w czasie panowania Żółtego Cesarza Huang–ti, kiedy to wiele rzeczy było możliwych, jak o tym świadczą zachowane legendy). Czas upływa im na myśleniu i medytacji w przeznaczonej do tego sali, muszą jednak od czasu do czasu jeść (zgodnie z klasyczną zasadą zachowania równowagi yin–yang). W jadalni stoi okrągły stół z pięcioma miseczkami, między którymi leży pięć pałeczek do ryżu, a na środku stołu znajduje się duża misa z niewyczerpalnym zapasem gotowanego ryżu - swego rodzaju róg (chińskiej) Amaltei . Etykieta mówi, że każdy filozof siada zawsze na tej samej poduszce, je zawsze ze swojej miseczki i może korzystać tylko z pałeczek znajdujących się po obu jej stronach. Filozofowie przestrzegają ustalonej etykiety jak prawdziwi konfucjaniści, na przykład nigdy nie sięgnęliby po ryż dłonią. Nie urządzają jednak żadnych zebrań, nie uznają hierarchii, nie posiadają żadnej informacji, dotyczącej spraw przyziemnych. Kiedy poczują głód to spontanicznie ruszają do jadalni.

Jak powinni oni odprawiać swój rytuał, aby nie umrzeć z głodu? Rozwiązanie powinno spełniać następujące warunki:

1. Filozof je tylko wtedy, gdy ma dwie pałeczki.
2. Dwóch filozofów nie może jednocześnie trzymać tej samej pałeczki.
3. Nie występuje blokada (sytuacja patowa). Może ona wystąpić np. wtedy, gdy wszyscy filozofowie podniosą lewe pałeczki i będą czekać na zwolnienie prawych.
4. Nikt nie może być zagłodzony. Oczywista z pozoru strategia, polegająca na poczekaniu, aż obie pałeczki będą wolne, może spowodować zagłodzenie dwóch filozofów (dlaczego?).
5. Żaden z filozofów nie zajmuje się tylko jedzeniem. Po zakończeniu posiłku każdy odkłada pałeczki i wraca do sali medytacji.
6. Filozofowie podnoszą i odkładają pałeczki po jednej naraz.
7. Nie można wyróżniać żadnego z filozofów (algorytmy ich działania powinny być takie same).

Jedno z rozwiązań wymaga zaangażowanie odźwiernego, pilnującego drzwi jadalni i pozwalającego przebywać w niej jednocześnie co najwyżej czterem filozofom. Dzięki temu co najmniej dwom filozofom, siedzącym przy stole, brakuje co najmniej jednego sąsiada, a zatem co najmniej jeden filozof może jeść (dlaczego?). Rozwiązanie problemu pięciu filozofów wygląda więc następująco. Na początku wszyscy filozofowie medytują w przeznaczonej do tego sali. Kiedy któryś z nich poczuje głód, wstaje i kieruje się do jadalni. Jeśli w jadalni jest mniej niż czterech filozofów, odźwierny przepuszcza go. W przeciwnym razie filozof siada u drzwi jadalni, czekając na znak odźwiernego. W jadalni filozof zajmuje swoje stałe miejsce i próbuje podnieść najpierw lewą pałeczkę. Jeśli jest ona używana przez lewego sąsiada, czeka na jej zwolnienie, po czym analogicznie stara się podnieść prawą. Mając obie pałeczki filozof zaczyna się posilać. Po nasyceniu głodu odkłada najpierw lewą, a następnie prawą pałeczkę, wstaje i bezszelestnie wraca na swoje miejsce w sali medytacji. Jeśli u progu jadalni są oczekujący filozofowie, odźwierny wskazuje na jednego z nich, pozwalając mu wejść do jadalni. Wybory odźwiernego są uczciwe, co znaczy, że żaden z oczekujących nie będzie pomijany w nieskończoność.

Edsger W. Dijkstra, który w 1971 roku opublikował tę starożytną historię, dostosował ją do realiów Zachodu, m.in. każąc filozofom jeść spaghetti dwoma widelcami. Próbując rozwiązać problem pięciu filozofów, wymyślił on mechanizm *semaforów*. Jak widać, znajomość mitologii (i to nie tylko grecko–rzymskiej) może być bardzo inspirująca.

Klasyczny semafor *S* jest zmienną całkowitą, przyjmującą wyłącznie wartości nieujemne, służącą do sterowania korzystaniem z sekcji krytycznych. Semafor przyjmujący tylko wartości 0 i 1 nazywamy semaforem binarnym. Na semaforze *S* można wykonywać dwie niepodzielne operacje. Próbę wejścia do sekcji krytycznej reprezentuje operacja **żądaj,** przy wyjściu z sekcji krytycznej należy wykonać operację **zwolnij**. Są one zdefiniowane następująco: (a) **żądaj**(S) – jeśli S>0, to S:=S–1 i wpuść proces do sekcji krytycznej, w przeciwnym razie wstrzymaj wykonywanie procesu; (b) **zwolnij**(S) – jeśli są jakieś procesy wstrzymane przez ten semafor, to

wznów jeden z nich (w sposób uczciwy, nie powodujący zagłodzenia żadnego z procesów, wstrzymanych przez ten semafor), w przeciwnym razie S:=S+1 (S:=1 dla semafora binarnego).

Napisz program, rozwiązujący problem ucztujących filozofów za pomocą semaforów. W języku Java w pakiecie java.util.concurrent jest zdefiniowana klasa Semaphore (patrz dokumentacja). Operacje **żądaj** i **zwolnij** noszą odpowiednio nazwy acquire i release.
Przedstaw filozofów jako procesy, sekcją krytyczną jest jedzenie, a zasobami dzielonymi są pałeczki do ryżu. Procesy są ponumerowane od 0 do 4, co odpowiada stałym miejscom filozofów przy stole i wykonują się współbieżnie. Użycie każdej pałeczki jest kontrolowane przez semafor binarny, a odźwierny jest reprezentowany przez semafor ogólny z wartością początkową 4.

2.
Przeanalizuj program z folderu zad2. Dwa wątki zwiększają 200 000 razy wspólny licznik (egzemplarz klasy `IntCell`) o 1. Po ich zakończeniu wartość licznika powinna oczywiście wynosić 400 000. Uruchom ten program kilka razy. Jak wyjaśnisz otrzymane wyniki?

3. a) Popraw powyższy program, wykorzystując mechanizm monitorów.
   b) Popraw powyższy program, wykorzystując mechanizm semaforów.

4.W folderze zad4 znajduje się aplikacja, rozwiązująca problem producent/konsument dla trzech producentów i konsumentów.
a) Czy użycie metody notify() zamiast notifyAll() byłoby bezpieczne w tym programie?
b) Co w tym programie można ulepszyć? Wystarczy opisanie ulepszeń (z uzasadnieniem).