

## Kolokwium z Programowania Współbieżnego 2019-02-02

Każde zadanie prosimy oddać na **osobnej kartce**. Prosimy pisać **jednostronnie**.

### Zadanie 1 (Semaforey, MP, 10p.)

W pewnym mieście na literę M. istnieje turystyczna linia tramwajowa, którą *Turyści* mogą podróżować za darmo. Na linii znajduje się  $P$  przystanków (dla ułatwienia przyjmiemy, że ich nazwy to  $0, 1, \dots, P-1$ ), po których jeden *Tramwaj* porusza się cyklicznie, tzn. po przystanku 0 jest przystanek 1, po 1 przystanek 2, a po  $P-1$  jest przystanek 0.

Tramwaj, *process Tramwaj()*, rozpoczyna pracę od przystanku *obecny* = 0, a następnie w nieskończonej pętli pozwala wsiąść/wysiąść pasażerom i przejeżdża do kolejnego przystanku, funkcja *void dojazd(int następny)*,

W każdym momencie Tramwaj może zawierać nie więcej niż  $N$  pasażerów, a każdy pasażer Tramwaju jest Turystą.

Turysta, *proces Turysta(int id, int start)*, każdego dnia odwiedza wiele atrakcji turystycznych. Każdy Turysta działa według tego samego schematu: rozpoczyna podróż na przystanku *start*, niedaleko hotelu, i do końca dnia, tj. w nieskończonej pętli, zwiedza pobliską atrakcję *void zwiedzanie(id)*; po zakończeniu zwiedzania, dowiaduje się ile przystanków chce przejechać aby dostać się do następnej atrakcji turystycznej *int kiedy\_wysiadam(int id)* i oczekuje na Tramwaj. Kiedy Turyście uda się wsiąść do Tramwaju to czeka, aż Tramwaj przejedzie wymaganą liczbę przystanków i wysiada, by rozpocząć zwiedzanie kolejnej atrakcji.

Wykorzystując semafory napisz procesy *process Tramwaj()* oraz *proces Turysta(int id, int start)* realizujące działanie powyższej linii tramwajowej.

Należy pamiętać, że Turyści nie lubią, gdy bez powodu przerywa się im podróż oraz, że kolejny przejazd rozpoczyna z przystanku na którym zakończyli poprzedni przejazd tramwajem.

### Zadanie 2 (Monitory, KSz, 10p.)

W kawiarni spotykają się na "randkach w ciemno" procesy należące do dwóch grup (zakładamy, że identyfikatory grup to 0 i 1). Każdy proces wie, do której grupy należy, natomiast liczności grup nie są znane i nie muszą być równe. W każdej randce udział bierze jeden proces z grupy o numerze 0 i jeden proces z grupy o numerze 1.

Kawiarnia dysponuje  $S$  identycznymi, dwuosobowymi stolikami, przy których odbywają się randki. Stoliki te ustawione są kółkiem wokół estrady, na której w niektóre dni odbywają się koncerty.

Proces przychodzący do kawiarni czeka na wolne miejsce przy stoliku. Miejsce jest wolne wtedy, gdy przy stoliku nikt nie siedzi albo siedzi przy nim jeden proces. Może to być proces należący do drugiej grupy, czekający na rozpoczęcie randki albo proces, który już zakończył randkę ale jeszcze nie zdążył odejść od stolika (musi on opuścić kawiarnię nawet wtedy, gdy drugie miejsce zostało zajęte przez nowy proces czekający na randkę). Po zajęciu miejsca proces czeka na pojawienie się partnera.

W momencie, gdy oba miejsca przy stoliku zostają zajęte przez procesy czekające na randkę, wzywają one kelnera, u którego zamawiają kawę.

Po wypiciu kawy procesy wzywają ponownie kelnera i czekają na rachunek. Po uregulowaniu rachunku odchodzą od stolika i opuszczają kawiarnię. Nie muszą tego robić jednocześnie.

Procesy działają zgodnie z algorytmem:

```
process process(int grupa) {  
    int i;  
    while (true) {  
        i = Kawiarnia.CzekamNaMiejscePartneraIKawe(grupa);  
        // pije kawę przy i-tym stoliku  
        Kawiarnia.CzekamNaRachunek(grupa, i);  
        // reguluje rachunek  
        Kawiarnia.ZwalniamMiejsce(grupa, i);  
    }  
}
```

W kawiarni zatrudniony jest tylko jeden kelner. Chodzi on "w kółko" i obsługuje kolejne stoliki w takiej kolejności, w jakiej są ustawione. Obsługa polega albo na podaniu kawy albo na dostarczeniu rachunku za nią. Jeśli kelner nie ma nic do roboty, usypia w oczekiwaniu na wezwanie.

Kelner realizuje następujący algorytm:

```
process kelner() {  
    int i=0;  
    while (true) {  
        i = Kawiarnia.KtoNastepny(i);  
        // podejście do i-tego stolika i podanie kawy albo albo dostarczenie rachunku  
        Kawiarnia.Gotowe(i);  
    }  
}
```

Napisz monitor *Kawiarnia* synchronizujący pracę kelnera obsługującego stoliki zgodnie z opisem powyżej.

**Powodzenia!**