

Úlohy na precvičenie – PDS 2019 – séria A

Úlohy riešte samostatne a podrobne. Celý postup zaznamenajte a komentujte. Odpovede zdôvodňujte celými vetami (väčšinou je otázok v úlohe viac!). Cieľom cvičenia nie je vyhľadanie riešenia, ale získanie skúseností so samostatným riešením úloh. Vo výnimočných situáciách použité citácie uveďte v riešení. Je slušné tiež uviesť mená osôb, s ktorými ste riešenie konzultovali (čo samozrejme je povolené, pokiaľ riešenie spíšete potom samostatne). Uprednostnite ručný zápis – umožní vám jednoducho pridávať ilustračné obrázky a nákresy.

Za každé správne a vyčerpávajúce riešenie (samozrejme aj s postupom) možno získať bod (ak nie je uvedené inak). Zlomky bodov možno získať aj za čiastočné riešenia. Riešenia tejto série je nutné doručiť do **21. 2. 2019, 14:25** (do začiatku prednášky). Pred týmto termínom je možné odovzdať riešenia na sekretariáte Ústavu informatiky (do priechinka Jirásek). Neskôr dodané riešenia a plagiáty nebudú opravované ani hodnotené. Problémy môžete konzultovať po prednáške resp. elektronickou poštou.

1. Ako by ste realizovali ošetrovanie vstupu do kritickej sekcie s aktívnym čakaním (spinlockom) pomocou atomickej inštrukcie `xchg`, ktorá navzájom vymení obsah registra s hodnotou v pamäti? (`void xchg(int *reg, int *mem)`). Zapište pomocou tejto inštrukcie tiež bezpečnú implementáciu zámku s pasívnym čakaním (mutexu) a semafora (funkcií `lock` a `unlock` resp. `signal` a `wait`).

2. Program používa semaforey `s1` a `s2` s počiatočným nastavením `s1.val = 0; s2.val = 1` a celočíselnú premennú `x` s počiatočnou hodnotou `x = 0`. Pozostáva z troch vláken: `{wait(s2); x=x+1; signal(s1); signal(s2)}`, `{wait(s2); x=x*2; signal(s2)}` a `{wait(s1); wait(s2); x=x*x; signal(s2)}`. Aké hodnoty môže nadobudnúť premenná `x` po ukončení výpočtu? Zdôvodnite.

3. Lamportov algoritmus na zabezpečenie vzájomného vylúčenia využíva pole `choosing` (slajdy z prenášky). Ukážte, že využitie poľa `choosing` je nevyhnutné pre správne fungovanie algoritmu. Teda ukážte, že pre algoritmus, v ktorom sú všetky riadky kódu pracujúce s poľom `choosing` odstránené, existuje také načasovanie vykonania jednotlivých príkazov algoritmu procesmi, v ktorom nebude dosiahnuté vzájomné vylúčenie (aspoň 2 procesy budú v tom istom čase v kritickej sekcii). Nájdite príklad s čo najmenším počtom procesov.