Programovanie paralélnych a distribuovaných systémov
Domáca úloha 4

Autor: Marián Kravec

Úloha 1

Ak požiadavku udn1 zmeníme z u.t unless u.h na u.t leads-to u.h tak by kód mal stále fungovať, respektíve časť os bude fungovať bez problémov (a keďže časť user nás netrápi tak ju môžme považovať, že je v poriadku). Dôvodom prečo bude fungovať je,že leads-to je silnejšia požiadavka ako unless ktorá hovorí to, že sa filozof nie len môže dostať z thinking na hungry ale vieme, že sa niekedy dostane.

Náš program dokonca môžeme zjednodušiť tým, že z neho odstránime request-token a špinavú vidličku budeme vždy posielať. Keďže vieme, že žiaden nebude večne v thinking stave tak nikdy nenastane situácia, že filozof bude držať vidličku a nikdy by nebol hladný čím by obmedzil svojich susedov.

Úloha 2

Ak požiadavky udn1 (u.t unless u.h) a udn2 (stable u.h) zmeníme za stable ¬u.e. V tomto prípade si myslím, že náš kód už fungovať správne nebude. Dôvodom je, že môže nastať situácia, že si filozof v hladnom stave vyžiada "špinavú"vidličku, dostane "čistúä následne prejde do thinikng stavu, v tomto stave teraz môže ostať teoreticky navždy a keďže drží "čistú"vidličku tak ju nechce pustiť čím obmedzuje svojho suseda.

Teoreticky riešenie tohto problému by mohlo byť, že by ak je filozof v stave thinking a má request-token od suseda tak mu pošle aj "čistú"vidličku. Toto sme pri pôvodných podmienkach nemuseli riešiť keďže filozof ktorý je thinking nikdy nemohol mať "čistú"vidličku (keďže na to aby mal "čistú"vidličku musel si ju vyžiadať na čo musel byť v stave hungry). Takéto posielanie "čistých"vidličiek v stave thinking rieši problém toho, že by filozof večne držal vidličku (v stave thinking) čím by obmedzoval suseda. Zároveň tým, že "čistú"vidličku pošle iba v stave thinking tak nenastáva problém, že by sa hungry filozof vzdal "čistej"vidličku čím by sa posunul ďalej v rade.