

1. (12,5) K1 Објасните шта је *At-Most-Once-Property*. Објасните зашто, када је испуњена та особина, критична референца има особине атомске акције. Дати два примера у којима су по два процеса.

2. (12,5) K1 Постоје три особе међу којима треба изабрати једну (*The Odd Person Wins Game*). Свака од тих особа поседује новчић који има две стране. Избор особе се одиграва тако што свака особа независно баца свој новчић. Уколико постоји особа којој је новчић пао на другу страну у односу на преостале особе, онда се та особа изабрава. Уколико све особе имају исто постављен новчић, поступак се понавља све док се не изабере једна. Особе након сваког бацања морају да знају да ли су изабране или не или се поступак понавља. Користећи семафоре написати програм који описује понашање особе. Ниједној особи не треба давати предност на основу њиховог идентификатора.

3. (12,5) K2 Монитор треба да регулише распоред уласка пацијената на преглед код једног лекара. Сваки регуларно заказани пацијент је пре приступања том монитору добио један од 12 хронолошки поређаних полусатних интервала за тај дан (прављење распореда није део задатка). Пацијент када дође на преглед (не мора да буде тачан) позива мониторинску процедуру *želim_da_se_pregledam* и том приликом доставља ID и почетак полусатног интервала у коме је њему заказан преглед. Ако је лекар заузет, пацијенти на чекању ће се поређати на основу хронолошког редоследа у распореду за тај дан. Лекар позива мониторинску процедуру *sledeći*, која враћа идентификатор следећег пацијента, када жели да прегледа следећег пацијента. Ако нема пацијената у том случају, лекар се поставља у стање чекања из кога излази када наиђе први пацијент. Написати овакав монитор користећи дисциплину *Signal and Wait* и условне променљиве код којих нема приоритетних редова чекања. Монитор не треба да садржи синхронизацију везану за крај прегледа.

4. (12,5) K2 Трајект за превоз аутомобила, камиона и аутобуса превози возила са обале на обалу. Трајект поседује N позиција које су линеарно постављене једна иза друге. Камион заузима три, аутобус две, а аутомобил једну позицију. Возила чекају на трајект у реду и на њега улазе једно по једно по редоследу у којем чекају у реду, док на трајекту има места. Када наредно возило у реду за трајект нема места да се укрца или када је трајект пун, трајект започиње превоз возила на другу обалу. На другој обали возила се искрцавају у редоследу супротном од редоследа у којем су се укрцала. Када се сва возила искрцају, празан трајект се враћа на почетну обалу. Користећи регионе написати програм који решава овај проблем.

5. (15) K3 Користећи синхрону размену порука написати дистрибуирано решење за проблем филозофа који ручавају (*The Dining Philosophers Problem*). Код дистрибуираног решења процеси филозофи комуницирају само са процесима виљушкама и процеси виљушке комуницирају само са процесима филозофима.

6. (15) K3 Постоји P произвођача и C потрошача који деле заједнички бафер капацитета B (*Atomic broadcast problem*). Произвођачи генеришу производ по производ на које чекају свих C потрошача. Сваки потрошач мора да прими производе у тачно оном редоследу у коме су произведени, мада различити потрошачи могу у исто време да узимају различите производе. Решити проблем користећи језик *C-Linda*, тако да се ни у ком тренутку у заједничком простору не нађе више од B производа.

Колоквијум 1 траје 1,5 сати.

Колоквијум 2 траје 1,5 сати.

Колоквијум 3 траје 2 сати.

Све остале комбинације трају 3 сати.

Напомена: На вежбанци назначити који део се ради (K1, K2, K3, K1+K2, K1+K3, K2+K3 или K1+K2+K3).