

1. (12.5) K1 Објасните основну разлику и разлоге за постојање те разлике између имплементација *bounded buffer*-а за случајеве 1 произвођач и 1 потрошач и M произвођача и N потрошача помоћу семафора. У складу са објашњењем, прикажите разлике и за случајеве M произвођача и један потрошач и 1 произвођач и N потрошача.

2. (12.5) K1 Постоји група од N филозофа који проводи свој живот тако што наизменично филозофирају, чекају на пиће, и пију (*The Drinking Philosophers Problem*). Филозофи су тако распоређени да су флаше са пићем постављене између суседних филозофа. У неком тренутку филозоф може да постане жедан. Жедном филозофу је потребно неколико суседних флаша да би направио коктел и почео да га пије. Избор пића зависи од тренутног расположења и може се разликовати од прилике до прилике. Када прикупи сва потребна пића филозоф започиње са њиховим испијањем које траје извесно време. Када се напије филозоф враћа флаше да и дуги уживају, а он прелази на филозофирање. Написати програм користећи семафоре, тако да филозоф када једном узме флашу не враћа је док не заврши са конзумирањем.

3. (12.5) K2 Специјалистичка лекарска ординација прима три категорије пацијената: ургентне, лежеће и заказане спољне пацијенте. Ургентни имају највиши приоритет, али морају да сачекају завршетак започетог прегледа пацијента било ког приоритета. Лежећи имају слабији приоритет, а заказани спољни најнижи приоритет. Унутар истих категорија пацијената са вишим приоритетима (ургентни и лежећи) треба да важи FIFO редослед, а унутар заказаних спољних пацијената важи редослед према времену заказивања, ако чекају. Направите монитор који обезбеђује одговарајући редослед пријема код лекара и тиме рад ординације. Мониторске процедуре за пацијенте су: *trazim_pregled* и *zavrsio_sam_pregled*. Ради поједностављења проблема, нема мониторинских процедура које позива лекар.

4. (12.5) K2 У гнезду постоји n птића и две родитељске птице (*The Hungry Birds Problem*). Птићи једу из заједничке посуде која прима F црвића. Сваки птић непрекидно једе из посуде по једног црвића, мало спава и поново се враћа да једе. Када посуда постане празна, птић који је испразнио посуду буди једног од родитеља. Родитељска птица може да крене у лов на црвиће само уколико у гнезду остане други родитељ.

Из лова се родитељ враћа тек када накупи F црвића које сипа у посуду. Родитељске птице осим тога што чувају птиће и иду у лов могу да напусте гнездо да би саме јеле, тако да у гнезду увек остане један родитељ. Користећи условне критичне регионе написати програм који симулира понашање птића и родитеља.

5. (15) K3 Реализовати филтерску мрежу за сортирање *integer* вредности која се састоји из процеса (елемената) који стапају улазне низове на следећи начин: Ако су два улазна низа сортирани, на излазу се добија сортирани низ, али у коме се не јављају елементи са истом вредношћу. Дакле, излазни сортирани низ из процеса (елемента) је краћи од збира дужина улазних низова, ако постоје исти елементи на улазу. Користити синхрони *message passing*. Максимално паралелизовати извршавање и израчунати колико је у том случају најмање процеса потребно, ако имамо 15 *integer*-а на улазу.

6. (15) K3 У студентском дому студенти могу да организују журке (*The Room Party Problem*). На журку могу да долазе студенти, на журци могу да пију и могу са журке да одлазе. На журци може бити произвољан број студената. Управник студентског дома долази у обилазак соба само у случају да у соби нема студената или у случају да их је више од 50. Када је управник у соби нови студенти не могу да улазе, али присутни могу да је напуштају. Управник излази из собе тек када сви студенти из ње изађу. Користећи C-Linda библиотеку написати програм за студенте и за управника.

Испити траје 3 сата.