

1. (12.5) K1 Петерсонов метод међусобног искључивања запосленим чекањем. Докажите да се заиста постиже међусобно искључивање, као и да се не може догодити *deadlock* без обзира на редослед и тренутке прекидања два процеса на улазу у критичну секцију.

2. (12.5) K1 Постоји тоалет капацитета N ($N > 1$) који могу да користе жене, мушкарци, деца и домар такав да важе следећа правила коришћења: у исто време се у тоалету не могу наћи и жене и мушкарци; деца могу да деле тоалет и са женама и са мушкарцима; дете може да буде у тоалету само ако се тамо налази барем једна жена или мушкарац; домар има ексклузивно право коришћења тоалета (*The Single Bathroom Problem*). Написати програм за жене, мушкараце, децу и домара који долазе до тоалета, користе га и напуштају га користећи семафоре. Може се претпоставити да су семафори фер.

3. (12.5) K2 Потребно је да N процеса деле два штампача. Пре употребе штампача, процес $P[i]$ позива мониторинску процедуру *request(printer)*, а процедура враћа идентитет штампача. По завршетку штампања, процес $P[i]$ ослобађа штампач са *release(printer)*. Дефинишите мониторинску инваријанту. Развити монитор који користи *Signal and Continue* дисциплину. Након тога, претпоставите да код *request*-а постоји додатни аргумент којим процес доставља свој приоритет. Модификујте монитор, тако да се штампачи добијају у складу са приоритетом. При истом приоритету треба да важи FCFS редослед.

4. (12.5) K2 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 је по једна флаша са пићем постављена 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 други уживају, а он прелази на филозофирање. Написати програм који симулира понашање филозофа користећи условне критичне регионе.

5. (15) K3 N процеса $node(i:1..N)$ представљају чворове модификованог бинарног уређеног стабла код кога је $node(1)$ корен. Сваки од њих има по две локалне променљиве *left* и *right* које представљају индексе левог, односно десног потомка, као и локалну променљиву *parent* која представља индекс родитеља (0 уколико такви не постоје). Сви подаци се налазе у листовима. Остали чворови представљају приступно стабло (слично Б+ стаблу) и чувају вредност од које су мањи сви подаци из њиховог левог подстабла. Сваки од ових процеса сме да комуницира само са својим потомцима и родитељем. Процес $node(0)$ шаље корену стабла вредност коју треба пронаћи у стаблу и њему треба вратити индекс чвора који садржи ту вредност. Може се извршавати више претрага истовремено. Реализовати процедуре за $node(0)$ и $node(i:1..N)$ у CSP -у.

6. (15) K3 На уласку у једну железничку станицу са једном улазном пругом и једним слепим колосеком десио се квар, па се на улазу направила колона међународних и домаћих возова. Квар је отклоњен и треба пуштати возове. Да би међународни возови мање каснили, они се пуштају први, по редоследу доласка. Пошто постоји само једна пруга и возови се не могу „претичати“, сви домаћи возови који су били испред међународних у колони се пребацују на слепи колосек који је довољно велики да сви возови могу да стану. Када сви међународни возови оду, пуштају се прво возови са слепог колосека, па онда преостали домаћи возови из колоне. Сама станица има N перона, тј. N возова истовремено могу да укрцавају и искрцавају путнике. Возови који у међувремену пристижу треба да буду опслужени, али новопристигли међународни немају приоритет у односу на возове који су испред њих у колони. Решити проблем користећи поштанске сандучиће. Водити рачуна о томе да су композиције возова тешке и да је потребно време да се воз помери са једног места на друго.

Исјинӣ ӣраје 3 са̄ӣа.

Напомена: На вежбанци назначити који део или делови се раде. Дозвољено је користити готове структуре података (листе, редове, стек, хеш мапе, стабла, ...).