

1. (20) **К** За *Readers-Writers* проблем направите решење прослеђивањем штафете/жетона (*passing the baton*) уз помоћ семафора, тако да сигнал код задовољава следеће услове: Процеси који читају могу да читају када већ има активних читача, само ако нема два или више закаснелих процеса који пишу, у супротном се закашњавају. Ако су у неком тренутку постојала бар два закаснела процеса који пишу, тада процеси који пишу имају приоритет у односу на процесе који читају, све док не нестану сви закаснели процеси који пишу. У осталим ситуацијама важи логика уобичајеног решења *Readers-Writers* проблема. Напишите општи сигнал код (без специфичности везаних за место у коду). Упутство: користите променљиву *dwflag* која је *true* ако су се појавила два закаснела процеса који пишу.

2. (20) **К** У свемиру постоји N небеских тела која међусобно интерагују (*N Body Gravitational Problem*). Користећи семафоре потребно је решити овај проблем користећи торбу послова за дохватање посла. Потребно је рализовати слереће: *Worker* (обавља израчунавање), *Bag* (обавља поделу посла) и *Collector* (обавља прикупњање резултата).

3. (20) **У** Линди реализовати лицитацију у којој постоји један процес *vođa_licitacije* и n процеса *učesnika_u_licitaciji* и један процес *tick* који ажурира време. Вођа лицитације треба да иницијализује торку са излицитираном вредношћу и да после датог временског интервала, ако је излицитирана вредност већа од резервисане вредности заврши лицитацију. Процеси учесници у лицитацији треба да се реализују тако да се не блокирају када се заврши лицитација.

4. (20) **У** неком забавишту постоји правило које каже да се на свака три детета мора наћи барем једна васпитачица (*The Child Care Problem*). Родитељ доводи једно или више деце у забавиште. Родитељ чека све док се не појави место у забавишту како би оставио децу и отишао. Васпитачица сме да напусти забавиште само уколико то не нарушава правило. Решити проблем користећи CSP.

1. (20) **К** За *Readers-Writers* проблем направите решење прослеђивањем штафете/жетона (*passing the baton*) уз помоћ семафора, тако да сигнал код задовољава следеће услове: Процеси који читају могу да читају када већ има активних читача, само ако нема два или више закаснелих процеса који пишу, у супротном се закашњавају. Ако су у неком тренутку постојала бар два закаснела процеса који пишу, тада процеси који пишу имају приоритет у односу на процесе који читају, све док не нестану сви закаснели процеси који пишу. У осталим ситуацијама важи логика уобичајеног решења *Readers-Writers* проблема. Напишите општи сигнал код (без специфичности везаних за место у коду). Упутство: користите променљиву *dwflag* која је *true* ако су се појавила два закаснела процеса који пишу.

2. (20) **К** У свемиру постоји N небеских тела која међусобно интерагују (*N Body Gravitational Problem*). Користећи семафоре потребно је решити овај проблем користећи торбу послова за дохватање посла. Потребно је рализовати слереће: *Worker* (обавља израчунавање), *Bag* (обавља поделу посла) и *Collector* (обавља прикупњање резултата).

3. (20) **У** Линди реализовати лицитацију у којој постоји један процес *vođa_licitacije* и n процеса *učesnika_u_licitaciji* и један процес *tick* који ажурира време. Вођа лицитације треба да иницијализује торку са излицитираном вредношћу и да после датог временског интервала, ако је излицитирана вредност већа од резервисане вредности заврши лицитацију. Процеси учесници у лицитацији треба да се реализују тако да се не блокирају када се заврши лицитација.

4. (20) **У** неком забавишту постоји правило које каже да се на свака три детета мора наћи барем једна васпитачица (*The Child Care Problem*). Родитељ доводи једно или више деце у забавиште. Родитељ чека све док се не појави место у забавишту како би оставио децу и отишао. Васпитачица сме да напусти забавиште само уколико то не нарушава правило. Решити проблем користећи CSP.