

1. (20) К Користећи расподељене бинарне семафоре и технику предаје штафетне палице решити проблем читалаца и писаца (*Readers - Writers Problem*).

2. (20) К Проблем изградње молекула воде (*The H₂O Problem*). Написати монитор са *signal and continue* дисциплином за решавање овог проблема, под следећим условима. Атоми водоника, када желе да направе молекул воде, позивају мониторску процедуру *hReady()*, атоми кисеоника позивају мониторску процедуру *oReady()*. Последњи пристигли атом треба да позове мониторску процедуру *makeWater()*, након чијег завршетка сва три атома треба да заврше своје одговарајуће *hReady()* и *oReady()* процедуре. Не сме бити изгладњивања.

3. (20) И Користећи активне мониторе решити проблем филозофа који ручавају (*The Dining Philosophers*). Филозофи могу да комуницирају искључиво са процесом координатором (централизовано решење). Обезбедити да филозоф који је пре затражио да једе пре и започиње са јелом. Написати код за филозофе и за процес координатор.

4. (20) И Посматра се семафор за регулацију саобраћаја на улици са једним пешачким прелазом. Када пешак стигне до пешачког прелаза, уколико је светло за пешаке зелено, он прелази улицу. Уколико је у моменту његовог доласка светло за пешаке црвено, он чека да се укључи зелено светло. Зелено светло за пешаке се укључује или након К секунди од појаве првог пешака који је затекао црвено светло, или након проласка С аутомобила од последњег активног зеленог светла за пешаке. Зелено светло за пешаке трајања Г секунди се пали само уколико је испуњен неки од наведених услова и барем један пешак чека. Потребно је у језику CSP написати процесе пешака, аутомобила и семафора, уколико је познато да у систему постоји N аутомобила и Т пешака. Доступна је функција *system_current_time* која враћа тренутно време система.

Исјиниј шраје з саиша.

Напомена: На вежбанци прецртати задатке који се не раде. Дозвољено је коришћење готових (неконкурентних) структура података (листа, стек, ...).

1. (20) К Користећи расподељене бинарне семафоре и технику предаје штафетне палице решити проблем читалаца и писаца (*Readers - Writers Problem*).

2. (20) К Проблем изградње молекула воде (*The H₂O Problem*). Написати монитор са *signal and continue* дисциплином за решавање овог проблема, под следећим условима. Атоми водоника, када желе да направе молекул воде, позивају мониторску процедуру *hReady()*, атоми кисеоника позивају мониторску процедуру *oReady()*. Последњи пристигли атом треба да позове мониторску процедуру *makeWater()*, након чијег завршетка сва три атома треба да заврше своје одговарајуће *hReady()* и *oReady()* процедуре. Не сме бити изгладњивања.

3. (20) И Користећи активне мониторе решити проблем филозофа који ручавају (*The Dining Philosophers*). Филозофи могу да комуницирају искључиво са процесом координатором (централизовано решење). Обезбедити да филозоф који је пре затражио да једе пре и започиње са јелом. Написати код за филозофе и за процес координатор.

4. (20) И Посматра се семафор за регулацију саобраћаја на улици са једним пешачким прелазом. Када пешак стигне до пешачког прелаза, уколико је светло за пешаке зелено, он прелази улицу. Уколико је у моменту његовог доласка светло за пешаке црвено, он чека да се укључи зелено светло. Зелено светло за пешаке се укључује или након К секунди од појаве првог пешака који је затекао црвено светло, или након проласка С аутомобила од последњег активног зеленог светла за пешаке. Зелено светло за пешаке трајања Г секунди се пали само уколико је испуњен неки од наведених услова и барем један пешак чека. Потребно је у језику CSP написати процесе пешака, аутомобила и семафора, уколико је познато да у систему постоји N аутомобила и Т пешака. Доступна је функција *system_current_time* која враћа тренутно време система.

Исјиниј шраје з саиша.

Напомена: На вежбанци прецртати задатке који се не раде. Дозвољено је коришћење готових (неконкурентних) структура података (листа, стек, ...).

