

1. (20) **К** Потребно је реализовати семафор који поред стандардних атомских операција *signal()* и *wait()* има и атомске операције *signal(n)* и *wait(n)* које интерну семафорску променљиву атомски увећава односно умањује за  $n$  уколико је то могуће, уколико није чека док не буде било могуће. Потребно је решити проблем користећи расподељене бинарне семафоре и технику предаје штафетне палице. Процес који је раније упутио *wait* захтев треба раније да обави своју операцију. Претпоставити да у било ком тренутку максимално  $N$  процеса може упутити захтев за приступ семафору.

2. (20) **К** Аутомобили који долазе са севера и југа морају да пређу реку преко моста (*One lane bridge problem*). На мосту постоји само једна возна трака. Значи, у било ком тренутку мостом може да прође један или више аутомобила који долазе из истог смера (али не и из супротног смера). Обезбедити да се смер саобраћаја мења сваки пут након што га пређе 10 аутомобила из истог смера, ако су за то време један или више аутомобила чекали да га пређу из супротног смера. Написати монитор са *signal and wait* дисциплином који решава дати проблем.

3. (20) **И** Филтерска мрежа за тражење минимума низа. Елемент мреже на своје улазе прима две низа целих бројева која се завршавају са вредношћу *EOS*. Елемент на једном излазу треба да генерише вредност која одговара минималној вредности из примљених низова, и након тога вредност *EOS*. Приказати филтерску мрежу која најбрже налази минимум 11 вредности.

4. (20) **И** Претпоставити да постоји  $N$  путника и једно возило на тобогану (*The roller coaster problem*). Путници се наизменично шетају по луна парку и возе на тобогану. Тобоган може да прими највише  $K$  путника при чему је  $K < N$ . Вожња тобоганом може да почне само уколико се сакупило тачно  $K$  путника. Користећи *C-Linda* написати програм који решава овај проблем.

Колоквијум + испитиј итраје 3 сати.

Колоквијум итраје 2 сати. Испитиј итраје 2 сати.

**Напомена:** На вежбанци назначити који део се ради (**К**, **И** или **К+И**).

1. (20) **К** Потребно је реализовати семафор који поред стандардних атомских операција *signal()* и *wait()* има и атомске операције *signal(n)* и *wait(n)* које интерну семафорску променљиву атомски увећава односно умањује за  $n$  уколико је то могуће, уколико није чека док не буде било могуће. Потребно је решити проблем користећи расподељене бинарне семафоре и технику предаје штафетне палице. Процес који је раније упутио *wait* захтев треба раније да обави своју операцију. Претпоставити да у било ком тренутку максимално  $N$  процеса може упутити захтев за приступ семафору.

2. (20) **К** Аутомобили који долазе са севера и југа морају да пређу реку преко моста (*One lane bridge problem*). На мосту постоји само једна возна трака. Значи, у било ком тренутку мостом може да прође један или више аутомобила који долазе из истог смера (али не и из супротног смера). Обезбедити да се смер саобраћаја мења сваки пут након што га пређе 10 аутомобила из истог смера, ако су за то време један или више аутомобила чекали да га пређу из супротног смера. Написати монитор са *signal and wait* дисциплином који решава дати проблем.

3. (20) **И** Филтерска мрежа за тражење минимума низа. Елемент мреже на своје улазе прима две низа целих бројева која се завршавају са вредношћу *EOS*. Елемент на једном излазу треба да генерише вредност која одговара минималној вредности из примљених низова, и након тога вредност *EOS*. Приказати филтерску мрежу која најбрже налази минимум 11 вредности.

4. (20) **И** Претпоставити да постоји  $N$  путника и једно возило на тобогану (*The roller coaster problem*). Путници се наизменично шетају по луна парку и возе на тобогану. Тобоган може да прими највише  $K$  путника при чему је  $K < N$ . Вожња тобоганом може да почне само уколико се сакупило тачно  $K$  путника. Користећи *C-Linda* написати програм који решава овај проблем.

Колоквијум + испитиј итраје 3 сати.

Колоквијум итраје 2 сати. Испитиј итраје 2 сати.

**Напомена:** На вежбанци назначити који део се ради (**К**, **И** или **К+И**).