

1. (20) **К** Напишите решење коришћењем *await* исказа решење за *ticket* алгоритам. Ако не постоји *fetch_and_add* инструкција, већ само *test_and_set*, реализуйте *fine grain* решење са запосленим чекањем за *ticket* алгоритам. Решење не треба да се бави проблемом инвалидације кеш меморија.

2. (20) **К** Користећи расподељене бинарне семафоре и технику предаје штафетне палице, написати функцију *zameni(vrednost)*, која за два узастопна позивајућа процеса мења њихове вредности, тј. први као повратну вредност треба да добије вредност другог позивајућег процеса, а други вредност првог; трећи добија вредност четвртог, а четврти трећег, итд.

3. (20) **И** У систему се извршавају три процеса F, G и H, при чему сваки процес има локални низ целих бројева. Сваки од низова је уређен неопадајуће. Најмање један исти број се појављује у сваком од ова три низа. Написати код за сваки од ова три процеса у којем три процеса међусобно комуницирају док сваки од њих не одреди најмању заједничку вредност. Поруке које се размењују могу да садрже само по један елемент низа. Користити асинхрону размену порука. Сваки процес треба да има идентичан код.

4. (20) **И** Користећи *C-Linda* прикажите рад радио-таксија. У решењу треба да постоје процеси корисника таксија, диспечера позива и самих таксија са радио станицама. Редослед при нормалној резервацији је следећи: корисник телефоном позове диспечера и искаже захтев за вожњу, диспечер пошаље захтев свим радио таксијима, таксији се јављају у периоду од n јединица времена, диспечер након истека тог времена објављује да пријављени такси најближи кориснику обавља превоз, јавља кориснику који такси га вози и на крају такси долази до корисника, вози га и завршава вожњу након неког времена. Уколико се након истека n јединица времена не јави ни један такси кориснику се јавља да нема слободних возила. Претпоставити да постоји метода *sleep(time)*.

Исцрпни њраје 3 саџа.

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

1. (20) **К** Напишите решење коришћењем *await* исказа решење за *ticket* алгоритам. Ако не постоји *fetch_and_add* инструкција, већ само *test_and_set*, реализуйте *fine grain* решење са запосленим чекањем за *ticket* алгоритам. Решење не треба да се бави проблемом инвалидације кеш меморија.

2. (20) **К** Користећи расподељене бинарне семафоре и технику предаје штафетне палице, написати функцију *zameni(vrednost)*, која за два узастопна позивајућа процеса мења њихове вредности, тј. први као повратну вредност треба да добије вредност другог позивајућег процеса, а други вредност првог; трећи добија вредност четвртог, а четврти трећег, итд.

3. (20) **И** У систему се извршавају три процеса F, G и H, при чему сваки процес има локални низ целих бројева. Сваки од низова је уређен неопадајуће. Најмање један исти број се појављује у сваком од ова три низа. Написати код за сваки од ова три процеса у којем три процеса међусобно комуницирају док сваки од њих не одреди најмању заједничку вредност. Поруке које се размењују могу да садрже само по један елемент низа. Користити асинхрону размену порука. Сваки процес треба да има идентичан код.

4. (20) **И** Користећи *C-Linda* прикажите рад радио-таксија. У решењу треба да постоје процеси корисника таксија, диспечера позива и самих таксија са радио станицама. Редослед при нормалној резервацији је следећи: корисник телефоном позове диспечера и искаже захтев за вожњу, диспечер пошаље захтев свим радио таксијима, таксији се јављају у периоду од n јединица времена, диспечер након истека тог времена објављује да пријављени такси најближи кориснику обавља превоз, јавља кориснику који такси га вози и на крају такси долази до корисника, вози га и завршава вожњу након неког времена. Уколико се након истека n јединица времена не јави ни један такси кориснику се јавља да нема слободних возила. Претпоставити да постоји метода *sleep(time)*.

Исцрпни њраје 3 саџа.

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