

1. (20) Написати и **објаснити** *Fine grain Ticket* алгоритам реализован помоћу *FA* операције.

Уколико би уместо *FA* постојала операција *TS (test and set)* која би недељиво обављала (

```
bool TS(bool lock) {  
    < bool initial = lock;  
    lock = true;  
    return initial; > };
```

да ли је могуће направити *Fine grain* решење и ако је могуће – направите га. **Објаснити** да ли је и зашто је то правична (*fair*) критична секција.

2. (20) Посматра се сто у мензи за којим може да седи највише N особа ($N > 2$). Особа узима храну, седа за сто (*sitAtTable()*), једе, устаје са стола (*leaveTable()*) и одлази. Потребно је обезбедити да ниједна особа не једе, тј. седи за столом сама. Такође, свака особа која је узела храну мора да седне за сто и једе. Користећи монитор са *signal and wait* дисциплином написати програм који решава овај проблем. Монитор има само две методе *sitAtTable()* и *leaveTable()*.

Колоквијум траје 120 минута.

1. (20) Написати и **објаснити** *Fine grain Ticket* алгоритам реализован помоћу *FA* операције.

Уколико би уместо *FA* постојала операција *TS (test and set)* која би недељиво обављала (

```
bool TS(bool lock) {  
    < bool initial = lock;  
    lock = true;  
    return initial; > };
```

да ли је могуће направити *Fine grain* решење и ако је могуће – направите га. **Објаснити** да ли је и зашто је то правична (*fair*) критична секција.

2. (20) Посматра се сто у мензи за којим може да седи највише N особа ($N > 2$). Особа узима храну, седа за сто (*sitAtTable()*), једе, устаје са стола (*leaveTable()*) и одлази. Потребно је обезбедити да ниједна особа не једе, тј. седи за столом сама. Такође, свака особа која је узела храну мора да седне за сто и једе. Користећи монитор са *signal and wait* дисциплином написати програм који решава овај проблем. Монитор има само две методе *sitAtTable()* и *leaveTable()*.

Колоквијум траје 120 минута.