

1. (20) **К** Угњеждени позиви мониторинских процедура и проблеми који се могу јавити том приликом. Илустровати на примеру монитора који позива мониторинске процедуре монитора који алоцира ресурсе онеме који ће најкраће да их користи.

2. (20) **К** Посматра се свечана сала у којој новоизабране судије полажу заклетву пред врховним судијом. У сали осим нових судија и врховног судије могу бити и посматрачи (пријатељи и породица). Судије се прво јављају службенику испред сале да су стигли, један по један, а затим улазе унутра и чекају да положе заклетву. Посматрачи могу неометано да улазе. Након што врховни судија уђе у салу, нико више не може да уђе, али посматрачи могу да изађу. Врховни судија чека да уђу сви који су приступили реду за јављање службенику, након чега почиње полагање заклетви. Сваки од судија полаже заклетву пред врховним судијом, добија сертификат и напушта салу. Када све судије положе заклетву, врховни судија може да напусти свечану салу. Након тога сви поново могу да улазе. Користећи условне критичне регионе, написати програме за судије, врховног судију и посматраче, као и део кода за иницијализацију.

3. (20) Реализовати прстен од  $n$  процеса, који комуницирају асинхроним прослеђивањем порука, а код кога процес 0 генерише случајан низ од  $m$  природних бројева у интервалу од 1 до 20.000 завршен са *sentinel* вредношћу EONAT (нпр 20.001). Преосталих  $n-1$  процеса одстрањују из низа све појаве првих  $n-1$  простих бројева. Сваки процес има по један различит прост број за који је задужен. Након одстрањивања простих бројева, модификован низ се дистрибуира свим процесима. Обезбедити максималну конкурентност.

4. (20) Помоћу CSP-а реализовати клијент-сервер апликацију у којој постоји стотину клијената  $K_i$  ( $i=1..100$ ), три сервера  $S_1$ ,  $S_2$  и  $S_3$  и један помоћни процес  $P$ . Клијенти шаљу захтеве процесу  $P$  који их наизменично прослеђује серверима, тако да захтев добија онај сервер који је до сада обрадио најмање захтева, а клијента обавештава о томе ком серверу је захтев прослеђен. У случају подједнаког броја захтева, приоритет има процес са мањим идентификатором (дакле,  $S_1 > S_2 > S_3$ ). Клијент прима одговор директно од сервера коме је прослеђен његов захтев.

*Исїиїї иїраје 3 саїа.*

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

1. (20) **К** Угњеждени позиви мониторинских процедура и проблеми који се могу јавити том приликом. Илустровати на примеру монитора који позива мониторинске процедуре монитора који алоцира ресурсе онеме који ће најкраће да их користи.

2. (20) **К** Посматра се свечана сала у којој новоизабране судије полажу заклетву пред врховним судијом. У сали осим нових судија и врховног судије могу бити и посматрачи (пријатељи и породица). Судије се прво јављају службенику испред сале да су стигли, један по један, а затим улазе унутра и чекају да положе заклетву. Посматрачи могу неометано да улазе. Након што врховни судија уђе у салу, нико више не може да уђе, али посматрачи могу да изађу. Врховни судија чека да уђу сви који су приступили реду за јављање службенику, након чега почиње полагање заклетви. Сваки од судија полаже заклетву пред врховним судијом, добија сертификат и напушта салу. Када све судије положе заклетву, врховни судија може да напусти свечану салу. Након тога сви поново могу да улазе. Користећи условне критичне регионе, написати програме за судије, врховног судију и посматраче, као и део кода за иницијализацију.

3. (20) Реализовати прстен од  $n$  процеса, који комуницирају асинхроним прослеђивањем порука, а код кога процес 0 генерише случајан низ од  $m$  природних бројева у интервалу од 1 до 20.000 завршен са *sentinel* вредношћу EONAT (нпр 20.001). Преосталих  $n-1$  процеса одстрањују из низа све појаве првих  $n-1$  простих бројева. Сваки процес има по један различит прост број за који је задужен. Након одстрањивања простих бројева, модификован низ се дистрибуира свим процесима. Обезбедити максималну конкурентност.

4. (20) Помоћу CSP-а реализовати клијент-сервер апликацију у којој постоји стотину клијената  $K_i$  ( $i=1..100$ ), три сервера  $S_1$ ,  $S_2$  и  $S_3$  и један помоћни процес  $P$ . Клијенти шаљу захтеве процесу  $P$  који их наизменично прослеђује серверима, тако да захтев добија онај сервер који је до сада обрадио најмање захтева, а клијента обавештава о томе ком серверу је захтев прослеђен. У случају подједнаког броја захтева, приоритет има процес са мањим идентификатором (дакле,  $S_1 > S_2 > S_3$ ). Клијент прима одговор директно од сервера коме је прослеђен његов захтев.

*Исїиїї иїраје 3 саїа.*

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