



НАПОМЕНЕ

1. Обавезно прочитати **СВЕ** напомене.
2. Предати само једно решење задатка. Решење снимити унутар home директоријума у директоријуму са Вашим бројем индекса.
3. Обавезно уписати име, презиме и број индекса у коментар на почетку датотеке. Непотписани задаци неће бити прегледани.
4. Решење које не може да се компајлира носи максимално **33% поена**.
5. Искоментарисати код и организовати га у функције. Није дозвољено коришћење глобалних променљивих. На крају програма ослободити сву динамички заузету меморију.
6. За време теста нема паузе. Излазак из учионице се сматра завршетком теста.
7. Асистенти за време теста дају информације везане искључиво за разумевање текста задатка.
8. Комплетно реализован и тачан C++ програм доноси **30 поена**.
9. Тест се снима.
10. Тест траје **135 минута**.

Задатак

Написати конкурентни програм који симулира round-robin распоређивање процеса у једнопроцесорском систему. У симулацији постоје 2 типа нити: нити које представљају процесе и демонска нит која представља специјални процес који распоређује друге процесе (распоредјивач). Како би се извршили, процеси морају извршити све своје наредбе. Распоредјивач распоређује на два нивоа: прво по редовима по round-robin принципу ("у круг"), а затим по FIFO принципу унутар сваког појединачног реда. За више детаља прочитати остатак текста.

Процес

Процеси су представљени нитима. Да би се извршио, процес мора да заузме процесор и изврши **све своје наредбе једну за другом без икаквог прекида**. Свака наредба траје између 100 и 300 милсекунди, где се трајање наредбе одређује као псеудо-случајни број. Уколико процес не може одмах да добије ресурсе неопходне за своје извршавање, он одлази да чека. У овој симулацији постоје **укупно 3 реда чекања** (редом, редови 0, 1 и 2) и овај број се приликом тестирања решења неће мењати. Процес који не може да настави своје извршавање **одлази да чека у ред чекања који му је унапред задат** (видети параметар `redni_broj_reda` први параметар методе `izvrsi`). Процес се увек увезује **на крај одабраног реда чекања** како би био испоштован FIFO поредак. Након што активни процес изврши све своје наредбе, он сигнализира нити која представља процес распоређивач да може да одабере наредни процес за извршавање.

Поставка задатка

Описану логику извршавања процеса имплементирати у методи `izvrsi` класе `Rasporedjivac`. Ова метода као параметре прима редом: редни број реда у којем ће процес чекати уколико има потребе за тим (број између 0 и 2), јединствени идентификатор процеса (ненегативан цео број) и број наредби које процес треба да изврши.



Исписи

- Након сваке извршене наредбе: `Proces <X> izvrzio naredbu <Y>.`, где је X идентификатор процеса, а Y редни број извршене наредбе.
- Након што први процес заузме процесор: `Rasporedjivanje pocinje od <Z> reda.`, где је Z број реда у који би процес који испишује ову линију стао да треба да чека.

Распоређивач

Распоређивач је представљен посебном нити која је активна током читавог трајања програма. Распоређивач се активира сваки пут када неки активни процес заврши извршавање свих својих наредби и потребно је распоредити нови процес на извршавање. Док активни процес извршава своје наредбе, распоређивач је неактиван и чека да процес заврши са извршавањем. Распоређивање се врши на два нивоа и то:

- **[Ниво 1]** ред из којег се бира наредни активни процес се бира по **round-robin, односно кружном алгоритму**. Нпр. први процес се бира из реда 0, други из реда 1, трећи из реда 2, четврти из реда 0, пети из реда 1, итд. **Round-robin распоређивање не мора обавезно почети од реда 0!** Ред од којег почиње распоређивање зависи од тога у који ред би стао први процес који је затекао процесор слободним. Нпр. ако је први стигао процес који је требало да стане у ред 2, бројање почиње од 2. У том случају, избор реда би ишао следећим редоследом: ред 2, ред 0, ред 1, ред 2, ред 0, ред 1, итд. Уколико је неки од редова празан када је ред да се из њега распореди процес, тај ред се прескаче и тражи се први наредни непразни ред по round-robin принципу.
- **[Ниво 2]** Када се одабере ред по горе наведеном принципу, бира се један од процеса из тог реда. Увек се бира процес који је хронолошки стигао први, односно процес који је на почетку реда чекања (FIFO) принцип. Одабрани процес се поставља за наредни активни процес.

Распоређивач не распоређује наредни активни процес у два случаја:

- **[Случај 1]** Ни у једном од 3 реда чекања нема процеса који чекају на процесор.
- **[Случај 2]** На самом почетку рада симулације када процесор заузима први процес који до њега стигне.

Поставка задатка

Описану логику распоређивања имплементирати у методи `rasporedi` класе `Rasporedjivac`.

Исписи

- Када распоређивач уђе у чекање: `Rasporedjivac cek...`
- Када се распоређивач активира: `Rasporedjivac aktiviran.`
- Уколико је распоређивач одредио који ће процес бити наредни активни: `Naredni proces se bira iz reda <X>`, где је X редни број реда из којег се бира активни процес.
- Уколико распоређивач није одредио који ће процес бити наредни активни:
`Ni jedan red nije rasporedjen na izvrshavanje jer nema procesa koji cekaju.`



Пример

У наставку је дат пример исписа добијен извршавањем програма. **Пример је оквирног карактера, ваш испис не мора и највероватније неће бити идентичан!**

```
Rasporedjivac ceka...
Kreiran proces 0 sa 1 naredbi. Zeli da stane u red 0.
Rasporedjivanje pocinje od 0 reda.
Kreiran proces 1 sa 1 naredbi. Zeli da stane u red 0.
Kreiran proces 3 sa 1 naredbi. Zeli da stane u red 1.
Kreiran proces 2 sa 3 naredbi. Zeli da stane u red 0.
Kreiran proces 4 sa 3 naredbi. Zeli da stane u red 1.
Kreiran proces 5 sa 2 naredbi. Zeli da stane u red 2.
Kreiran proces 7 sa 1 naredbi. Zeli da stane u red 2.
Kreiran proces 8 sa 2 naredbi. Zeli da stane u red 2.
Kreiran proces 6 sa 1 naredbi. Zeli da stane u red 2.
Proces 0 izvrzio naredbu 1.
Proces 0 zavrzio izvrshavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 1
Rasporedjivac ceka...
Proces 3 izvrzio naredbu 1.
Proces 3 zavrzio izvrshavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 2
Rasporedjivac ceka...
Proces 5 izvrzio naredbu 1.
Proces 5 izvrzio naredbu 2.
Proces 5 zavrzio izvrshavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 0
Rasporedjivac ceka...
Proces 1 izvrzio naredbu 1.
Proces 1 zavrzio izvrshavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 1
Rasporedjivac ceka...
Proces 4 izvrzio naredbu 1.
Proces 4 izvrzio naredbu 2.
Proces 4 izvrzio naredbu 3.
Proces 4 zavrzio izvrshavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 2
Rasporedjivac ceka...
Proces 7 izvrzio naredbu 1.
Proces 7 zavrzio izvrshavanje.
Rasporedjivac aktiviran.
```



Naredni proces se bira iz reda 0
Rasporedjivac ceka...
Proces 2 izvršio naredbu 1.
Proces 2 izvršio naredbu 2.
Proces 2 izvršio naredbu 3.
Proces 2 završio izvršavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 2
Rasporedjivac ceka...
Proces 8 izvršio naredbu 1.
Proces 8 izvršio naredbu 2.
Proces 8 završio izvršavanje.
Rasporedjivac aktiviran.
Naredni proces se bira iz reda 2
Rasporedjivac ceka...
Proces 6 izvršio naredbu 1.
Proces 6 završio izvršavanje.
Rasporedjivac aktiviran.
Ni jedan red nije rasporedjen na izvršavanje jer nema procesa koji cekaju.
Rasporedjivac ceka...