НАПОМЕНЕ

1. Обавезно прочитати **СВЕ** напомене.
2. Решење снимити под именом **`main.cpp`**. Ово је једина датотека која ће бити прегледана.
3. Обавезно уписати име, презиме и број индекса у коментар на почетку датотеке.
4. Решење **мора** да се компајлира.
5. На одбрану донети **потпуно тачно** решење.
6. Потребно је имплементирати **само** `izvrsiOperacije` методу како је објашњено у задатку.
7. Закоментарисане позиве фунција у `main`-у откоментарисати један по један како имплементирате задатак да бисте тестирали појединачне функционалности које имплементирате.

Задатак

Написати конкурентни Ц++ програм који симулира извршавање више процеса на једнопроцесорском рачунару, при чему је прекључивање регулисано применом кооперативног мултитаскинга. Процеси међусобно размењују текстуалне поруке и тиме се симулира интерпроцесна комуникација.

Процеси и операције

Процеси се састоје од више операција, које се извршавају једна за другом. Операција може бити једна од следећих:

- Нормална
- Препусти
- Пошаљи
- Прими.

Овај тим операције је дефинисан енумерацијом `TipOperacije`.
Нормална операција се симулира трајањем од 200 мс.
Препусти је специјална операција којом процес отпушта процесор и препушта га другим процесима (више о овоме у наставку текста).
Пошаљи и Прими су операције којима се обавља интерпроцесна комуникација (више о овоме у наставку текста).

Интерпроцесна комуникација

Интерпроцесна комуникација која се имплементира у овом задатку је таква да постоји једно сандуче за поруке за све поруке које се размењују у систему. Сви процеси могу вршити упис у ово сандуче (што чине извршавањем операције Пошаљи) и сви процеси могу читати садржај овог сандучета (извршењем

операције Прими). Садржај који се размењује међу процесима, односно садржај сандучета је обичан стринг. Извршавањем операције Пошаљи процес шаље поруку у сандуче. Операција Пошаљи има параметар садржај - што је стринг који се уписује у сандуче. Уколико сандуче већ садржи неки текст, онда се постојећи садржај брише и нови садржај се уписује у сандуче (сандуче увек садржи само најновију поруку). Након слања поруке, процес наставља да се извршава (не чека да неки други процес прими ту поруку).
Када процес извршава операцију Прими онда извршење зависи од тога да ли сандуче садржи неку поруку или не.

1. Уколико сандуче садржи поруку онда процесс чита садржај сандучета и обавезно исписује на екран садржај који је прочитан. Он не мења садржај сандучета, само га чита. Процес који је извршио операцију Прими након тога нормално наставља извршавање својих операције (не долази до прекључивања).
2. Уколико дође до извршења операције Прими а сандуче је празно, онда процес који је извршио ту операцију одлази у стање чекања, чекајући да се појави неки садржај у сандучету. Док је он блокиран, чекајући да се појави садржај у сандучету, остали процеси могу да раде, тако да је у том моменту потребно иницирати прекључивање. Након што се појави садржај у сандучету, процес прелази из стања чекања у ред спремних процеса и када процесор крене да извшрава тај процес, он може прочитати ту поруку.

Обратити пажњу на то да може више процеса да оде у стање чекања на поруку у сандучету пре него што се нешто појави у сандучету. Увек је потребно све процесе који чекају обавестити да је сандуче добило неки садржај.

Функција нити: fn

Процеси су представљени нитима које извршавају функцију fn. Процеси се извршавају у оквиру тест сценарија који су већ дати у задатку. Тест сценарија су функције чији позив почиње са test сваки од њих покреће неколико нитии како би приказао како се урађени задатак понаша у различитим свенаријима.
Следи кратак опис параметара функције fn. Функција прима референцу на објекат класе OC, редни број нити (параметар ид) и референцу на структуру Процес. Структура Процес описује процес који се извршава - процес има дефинисано време чекања, прослеђено у пољу pauza, тако да су операције које они покрећу одложене у времену. Поље öperacijeзадаје вектор операција које процес треба да изврши.
Функција fn, односно процеси који се извршавају, позивају методу ïzvrsiOperacijenад дељеним објектом os. Имплементација ове методе класе OS представља део пројектног задатка. Ова метода прима референцу на структуру Процес који треба да изврши. Операције треба да се извршавају једна за другом, од прве у том вектору до последње.
Сви елементи вектора операција имају дефинисано поље tip које прослеђује тип операције која треба да се изврши. Када је елемент вектора операција која има тип ”POSALJI” онда на тој позицији вектора и поље sadrzajйма дефинисану вредност. За остале типове операција то поље нема значење.

Ограничења

- За СВАКУ операцију која се извршава треба исписати која операција је у питању и ИД процеса.
- Приликом извршења операције ПРИМИ треба исписати садржај поруке која је прочитана из сандучета.
- Када се процес изврши (све операције су извршене) - треба исписати да је процес завршен.
- Поруке морају бити у формату датом у примеру.

Примери извршавања

Приказан је пример извршавања права два теста. Пример је као референтни, испис који добијете не мора бити апсолутно исти као онај приказан овде.

Тест 1

```
TEST: prekljucivanje prilikom izvrsenja operacije PREPUSTI
********************
[proces][0] je pokrenut, pauzira: 0 milisekundi
[proces][0] Izvrsenje operacije NORMALNA
[proces][1] je pokrenut, pauzira: 100 milisekundi
[proces][0] Izvrsenje operacije PREPUSTI
[proces][1] Izvrsenje operacije NORMALNA
[proces][1] Izvrsenje operacije NORMALNA
[proces][1] Izvrsenje operacije PREPUSTI
[proces][0] Izvrsenje operacije NORMALNA
[proces][1] Zavrsen
[proces][0] Zavrsen
```

Тест 2

```
TEST: razmena poruke izmedju 2 procesa, proces koji prima poruku nalazi spremnu poruku\\
u sanducetu
********************
[proces][0] je pokrenut, pauzira: 0 milisekundi
[proces][0] Izvrsenje operacije POSALJI
[proces][1] je pokrenut, pauzira: 100 milisekundi
[proces][0] Izvrsenje operacije PREPUSTI
[proces][1] Izvrsenje operacije PRIMI
[proces][0] Zavrsen[proces][
1] Primljena poruka: Zdravo
[proces][1] Zavrsen
```

Тест 3

```
TEST: razmena poruke izmedju 2 procesa, proces koji prima poruku nalazi spremnu\\
poruku u sanducetu
********************
[proces][0] je pokrenut, pauzira: 0 milisekundi
[proces][0] Izvrsenje operacije POSALJI
[proces][1] je pokrenut, pauzira: 100 milisekundi
[proces][0] Izvrsenje operacije PREPUSTI
[proces][1] Izvrsenje operacije PRIMI
[proces][0] Zavrsen
[proces][1] Primljena poruka: Zdravo
[proces][1] Zavrsen
```

Тест 5

```
TEST: razmena poruke izmedju 3 procesa, proces koji prima poruku nalazi spremnu poruku\\
u sanducetu
********************
[proces][0] je pokrenut, pauzira: 0 milisekundi
[proces][0] Izvrsenje operacije POSALJI
[proces][[proces][0] Izvrsenje operacije PREPUSTI1
] je pokrenut, pauzira: 100 milisekundi
[proces][2] je pokrenut, pauzira: 200 milisekundi
[proces][0] Zavrsen
[proces][1] Izvrsenje operacije POSALJI
[proces][1] Izvrsenje operacije PREPUSTI
[proces][1] Zavrsen
[proces][2] Izvrsenje operacije PRIMI
[proces][2] Primljena poruka: Nova_poruka
[proces][2] Zavrsen
```

Тест 6

```
TEST: razmena poruke izmedju 2 procesa, proces koji prima poruku mora da ceka na poruku
********************
[proces][0] je pokrenut, pauzira: 0 milisekundi
[proces][0] Izvrsenje operacije PRIMI
[proces][1] je pokrenut, pauzira: 100 milisekundi
[proces][1] Izvrsenje operacije POSALJI
[proces][1] Izvrsenje operacije PREPUSTI
[proces][0] Primljena poruka: Zdravo
[proces][0] Izvrsenje operacije NORMALNA
[proces][1] Zavrsen
[proces][0] Zavrsen
```

Тест 7

```
TEST: razmena poruke izmedju 3 procesa, proces koji prima poruku mora da ceka na poruku
********************
[proces][0] je pokrenut, pauzira: 0 milisekundi
[proces][0] Izvrsenje operacije PRIMI
[proces][1] je pokrenut, pauzira: 100 milisekundi
[proces][2] je pokrenut, pauzira: 200 milisekundi
[proces][1] Izvrsenje operacije PRIMI
[proces][2] Izvrsenje operacije POSALJI
[proces][2] Izvrsenje operacije PREPUSTI
[proces][0] Primljena poruka: Vise_primalaca
[proces][1] Primljena poruka: Vise_primalaca
[proces][1] Izvrsenje operacije NORMALNA
[proces][2] Zavrsen
[proces][0] Izvrsenje operacije NORMALNA
[proces][1] Zavrsen
[proces][0] Zavrsen
```