

OpenMP/MPI

Рачунарски системи високих перформанси

Петар Трифуновић

Вељко Петровић

Факултет техничких наука
Универзитет у Новом Саду

Рачунарске вежбе, Зимски семестар 2022/2023.



1. Када имате много обавеза одједном, најчешће:
 - a. Радите све сами у исто време
 - b. Тражите од пријатеља да Вам помогну
 - c. Радите све сами, једно по једно

2. У колективу Вам највише одговара:
- a. Када постоји вођа тима
 - b. Када су сви равноправни
 - c. Избегавате колективе

3. Храну највише волите да конзумирате:
- a. Виљушком, класично
 - b. Чачкалицом — нова чачкалица за сваку врсту хране
 - c. Чачкалицом — једна чачкалица за све

4. Од понуђених ствари које нуди данашња технологија, највише волите да користите:
- a. Дељене документе, Google Drive, Google Docs...
 - b. Апликације за дописивање
 - c. Не волите да користите технологију

Квиз — резултати

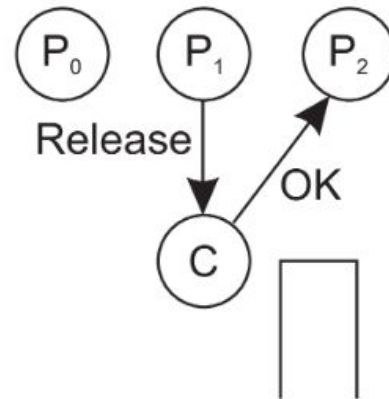
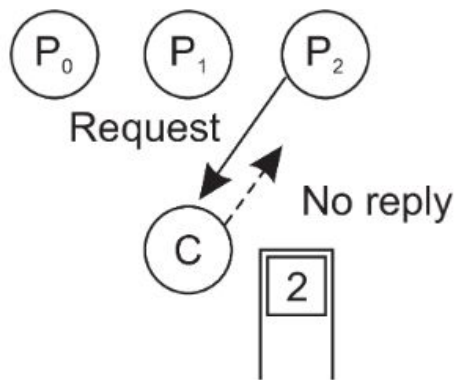
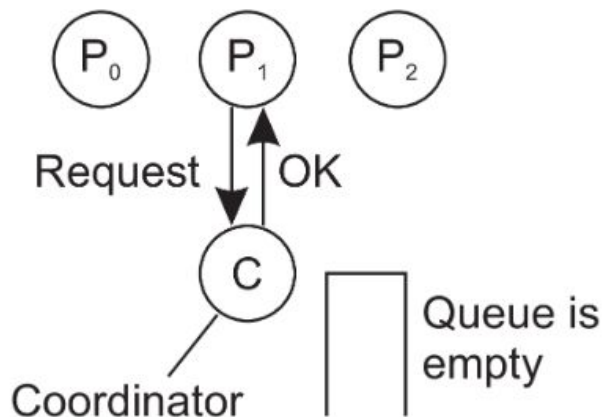
- **Највише одговора под а — Ви сте *OpenMP*:**
 - Волите да поделите посао на блиске јединке које раде истовремено
 - Сматрате да је најефикасније комуницирати путем дељених медијума, где сви могу видети поруку одмах, уместо да шаљете појединачно
 - Волите да се зна ко је главни
- **Највише одговора под б — Ви сте *MPI*:**
 - Волите да распоредите посао по сваку цену, макар и јединкама које нису тако блиске
 - Сматрате да је најбоље постарати се да свако добије своју поруку
 - Не обраћате пажњу на концепт вође, сви су равноправни, плен се дели
- **Највише одговора под с — Ви сте секвенцијално програмирање!**
Стидите се!

Имплементација дистрибуираних алгоритама

- Подсетимо се неких од алгоритама са предавања из предмета ПДАСП:
 - Централизован алгоритам заснован на дозволи
 - Лампортови логички сатови

Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи

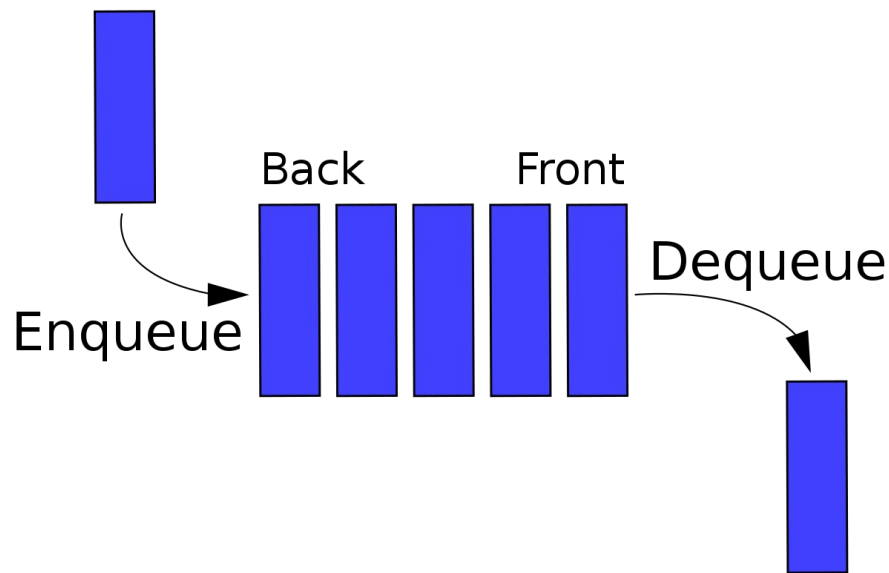
- **Координатор** — обезбеђује синхронизацију и јединствен приступ дељеном ресурсу
- Сваки учесник тражи дозволу за приступ од координатора



Преузето из материјала са предавања за предмет *Паралелни и дистрибуирани алгоритми и структуре података*

Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи

- Први корак — имплементирати ред чекања у C-у
- У позадини реда може бити најобичнији низ



Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи



- Други корак — моделовати логику проблема
- Питања везана за моделовање проблема:
 - Како процес зна да ли је координатор?
 - Како ће координатора разграничити између захтевања и отпуштања дозволе?
 - Како ће координатор знати да ли сме да додели дозволу?
 - Како ће координатор знати коме следећем да додели дозволу након што је претходни процес отпусти?

Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи



- Други корак — моделовати логику проблема
- Одговори:
 - Како процес зна да ли је координатор:
 - На основу ранка
 - Како ће координатора разграничити између захтевања и отпуштања дозволе?
 - На основу вредности примљене поруке
 - Како ће координатор знати да ли сме да додели дозволу?
 - На основу вредности посебне *boolean (int)* променљиве
 - Како ће координатор знати коме следећем да додели дозволу након што је претходни процес отпусти?
 - На основу садржаја реда чекања; дозволу добија процес који је најдуже у реду

Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи



- Трећи корак — моделовати комуникацију
- Питања везана за моделовање комуникације:
 - Од кога и када координатор прима поруку?
 - Када координатор шаље поруку?
 - Када радници примају поруку?
 - Када радници шаљу поруку?
 - Да ли се ради о комуникацији један-на-један, или о колективној?
 - Да ли користити блокирајуће или неблокирајуће позиве?

Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи



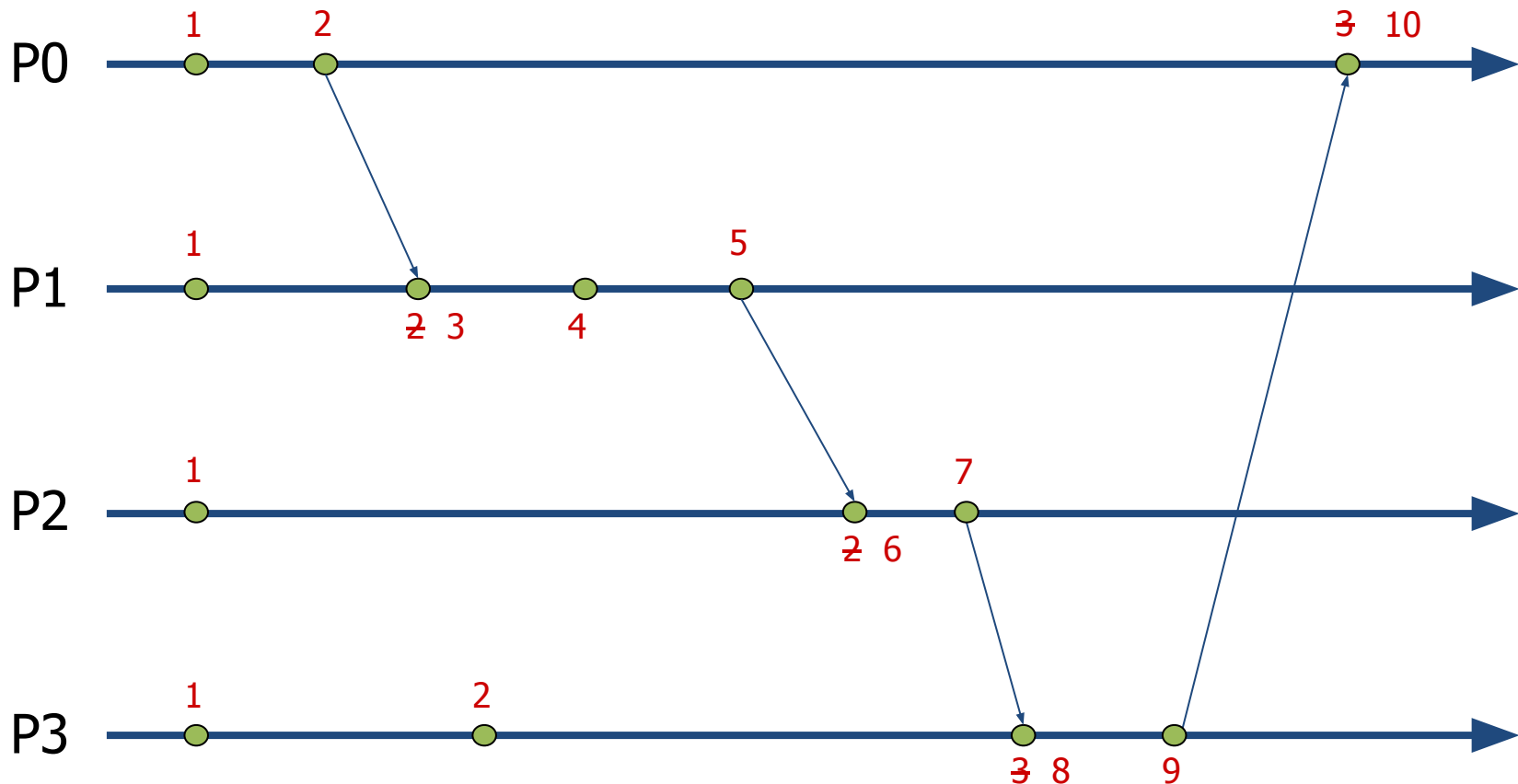
- Трећи корак — моделовати комуникацију
- Одговори:
 - Од кога и када координатор прима поруку?
 - Од било ког процеса (*MPI_ANY_SOURCE*) на почетку сваке итерације своје бесконачне петље
 - Када координатор шаље поруку?
 - Када додели некое дозволу
 - Када радници примају поруку?
 - Након што пошаљу захтев
 - Када радници шаљу поруку?
 - На почетку своје итерације и када отпушта дозволу
 - Да ли се ради о комуникацији један-на-један, или о колективној?
 - Један на један
 - Да ли користити блокирајуће или неблокирајуће позиве?
 - Или само блокирајуће, или оба

Имплементација дистрибуираних алгоритама — централизован алгоритам заснован на дозволи

- Решење: директоријум *centralizovana_dozvola*

- **Lamportovi logički satovi – algoritam :**
 1. Proces P_i , pre nego što izvrši bilo koji događaj (slanje ili prijem poruke, interni događaj), inkrementira C_i za 1
 2. Kada se poruka m šalje od strane procesa P_i , poruka dobija vremenski otisak $ts(m) = C_i$
 3. Kada se poruka m primi od strane procesa P_j , P_j podešava svoj lokalni brojač $C_j = \max\{C_j, ts(m)\}$; onda izvršava korak 1 i potom prosledi poruku m aplikaciji

Лампортови логички сатови — пример за имплементацију



Лампортови логички сатови — пример за имплементацију

- Решење: директоријум *lamportovi_logicki_satovi*

Задатак — обрада слике

- Најједноставнија репрезентација слике:
 - Заглавље са важним информацијама (величина слике, број канала по пикселу...)
 - Пиксели — за *RGB* слике, ово је матрица пиксела, где сваки пиксел садржи најчешће три бајта
 - Формати — *.bmp*, *.ppm*...

Задатак — обрада слике

- Циљ задатка:
 - Ротирати пикселе слике улево за одређен број позиција
 - “Унутрашњи” пиксели се померају на леву страну
 - Пиксели “по ободу”, који након померања испадају ван димензија слике, прелазе на десну страну



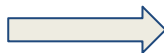
Задатак — обрада слике

- Детаљи имплементације:
 - Само један процес (*root*) учитава слику
 - Затим шаље осталим процесима информације о слици, па распоређује по свим процесима, укључујући и себе, одговарајућ број редова слике
 - Процеси обрађују своје редове и шаљу их назад *root*-у
 - *Root* чува слику на фајл систему рачунара



Задатак — обрада слике

- Детаљи имплементације:
 - Размотрити могућност хибридне имплементације



Задатак — обрада слике

- Решење: директоријум *obrada_slike*