

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
Кафедра комп'ютерних системи та мереж

# **ПАРАЛЕЛЬНІ ТА РОЗПОДІЛЕНІ ОБЧИСЛЕННЯ**

Методичні вказівки до виконання  
лабораторних робіт кредитного модуля  
«Паралельні та розподілені обчислення»  
для студентів напрямку 123  
«Комп'ютерна інженерія»

**КИЇВ - 2019**

Метою виконання лабораторних робіт з дисципліни «Паралельні та розподілені обчислення» є закріплення теоретичних знань, вмінь та навичок розробки і аналізу паралельних алгоритмів, розробки та налагодження програм для паралельних комп'ютерних систем (ПКС), отримання практичних навичок по роботі з паралельними мовами (бібліотеками) програмування.

Лабораторні роботи включають два цикли:

- цикл 1 «Програмування для комп'ютерних систем зі **спільною пам'яттю ПКС СП**» (1-6 роботи)
- цикл 2 «Програмування для комп'ютерних систем з **локальною пам'яттю ПКС ЛП**» (7-8 роботи).

## **ЗАВДАННЯ НА ЛАБОРАТОРНІ РОБОТИ**

Завдання на кожну лабораторну роботу видається викладачем (Додаток Б.Таблиця 3). Завдання включає:

- структуру ПКС (ПКС СП або ПКС ЛП)
- математичну задачу
- мову (або бібліотеку) паралельного програмування
- засоби організації взаємодії процесів (потоків).

Виконання лабораторній роботи пов'язане зі створенням **паралельної програми** і включає розробку і аналіз алгоритму рішення вхідної задачі, створення і налагодження програми.

Звіт по лабораторній роботі (протокол) включає опис всіх етапів розробки:

1. Паралельного математичного алгоритму,
2. Алгоритмів всіх паралельних потоків (процесів),
3. Структурної схеми взаємодії потоків,
4. Програми.

## **ЛАБОРАТОРНІ РОБОТИ ЦИКЛУ 1**

### **«Програмування для комп'ютерних систем зі спільною пам'яттю»**

Лаб. роботи 1-5 пов'язані з програмуванням для ПКС зі спільною пам'яттю. Кількість процесорів 2-4. На рис 1. наведений приклад структурної схеми ПКС зі СП. Включає 4 процесори (або ядра) та від одного до чотирьох пристроїв введення – виведення (ПВВ).

Для кожної лабораторної роботи необхідно:

- розробити паралельний алгоритм рішення математичної задачі;
- описати алгоритм кожного процесу (поток) ( $T_1$ - $T_2$  для лаб 1) ( $T_1$  –  $T_4$  для лаб 2-6) з визначенням критичних ділянок (КД) і точок синхронізації (ТС) ( $W_{ij}$ ,  $S_{ij}$ );
- розробити структурну схему взаємодії задач, де визначити засоби взаємодії потоків (процесів) (Див. Додаток А))
- розробити програму
- виконати налагодження програми;
- отримати *правильні* результати обчислень.
- за допомогою Диспетчеру задач Windows проконтролювати завантаження ядер процесору.

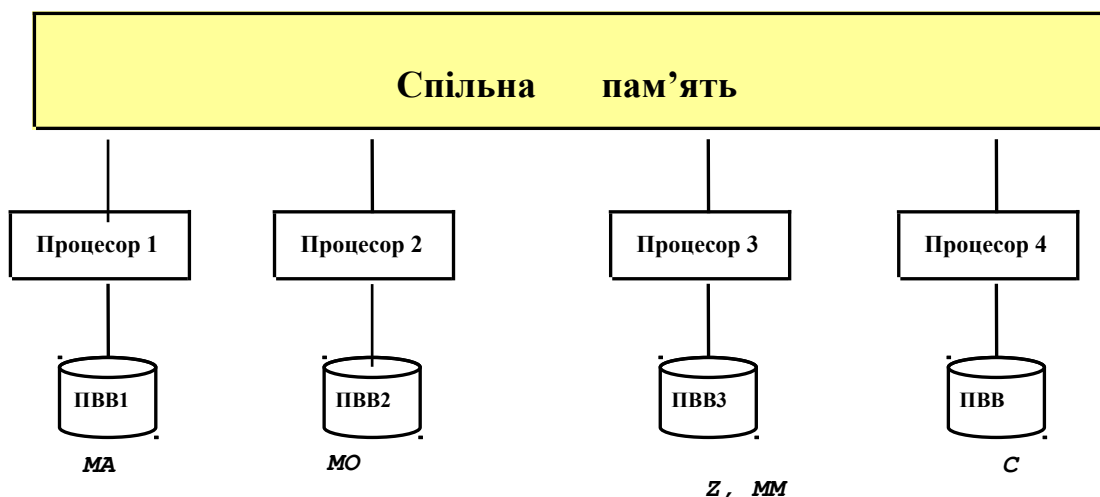


Рис. 1. Структура ПКС зі спільною пам'яттю (ПКС СП)

Лабораторні роботи першого циклу містять три види робіт

- використання засобів низького рівня: семафори, мютекси, події, критичних секцій (роботи 1-3)
- використання моніторів (роботи 5-6)
- бібліотека OpenMP (робота 4)

## ЛАБОРАТОРНА РОБОТА 1

### Ада.Семафори

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** Ада

**Засоби організації взаємодії процесів:** семафори мови Ада з пакета

`Ada.Synchronous_Task_Control`.

**Література:** [ 1, Розділ 6, с. 156-162.], [ 2, Цикл 3, с. 63 -68]

## ЛАБОРАТОРНА РОБОТА 2

### Win32. Семафори, мютекси, події, критичні секції

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** за вибором (С, Ада, Java)

**Засоби організації взаємодії процесів:** семафори, мютекси, події, критичні секції бібліотеки WinAPI

**Література:** [ 1, Розділ 6, с. 162-165.], [ 2, Цикли 3,4, с. 69 -75, 90-92, 117-121]

## ЛАБОРАТОРНА РОБОТА 3

### С#. Семафори, мютекси, події, критичні секції

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** C#

**Засоби організації взаємодії процесів:** семафори, мютекси, події, критичні секції мови C#

**Література:** [ 2, Цикли 3,4, с. 74 -77, 95-96, 121-122]

## **ЛАБОРАТОРНА РОБОТА 4** **OpenMP. Бар'єри, критичні секції**

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** за вибором (C, C++, Фортран)

**Засоби організації взаємодії процесів:** бар'єри, критичні секції OpenMP

**Література:** [ 1, Цикл 3, с. 28 -32]

## **ЛАБОРАТОРНА РОБОТА 5** **Ада. Монітори (Захищені модулі)**

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** Ада

**Засоби організації взаємодії процесів:** монітори (захищені модулі мови Ада).

**Література:** [ 1, Розділ 6, с. 165-170.], [ 2, Цикл 5, с. 104 -107]

## **ЛАБОРАТОРНА РОБОТА 6** **Java. Монітори**

**Мета роботи:** розробка програми для ПКС зі СП

**Мова програмування:** Java

**Засоби організації взаємодії процесів:** монітори мови Java

**Література:** [ 1, Розділ 6, с. 170-174.], [ 2, Цикл 5, с. 108 -112]

## **ЛАБОРАТОРНІ РОБОТИ ЦИКЛУ 2**

### **«Програмування для комп'ютерних систем з локальною пам'яттю»**

Лаб. роботи циклу два (роботи 7 і 8) зв'язані з програмуванням для ПКС з локальною (розподіленою пам'яттю). Структура ПКС з ЛП ( лінійна структура) наведений на рис. 2.) .

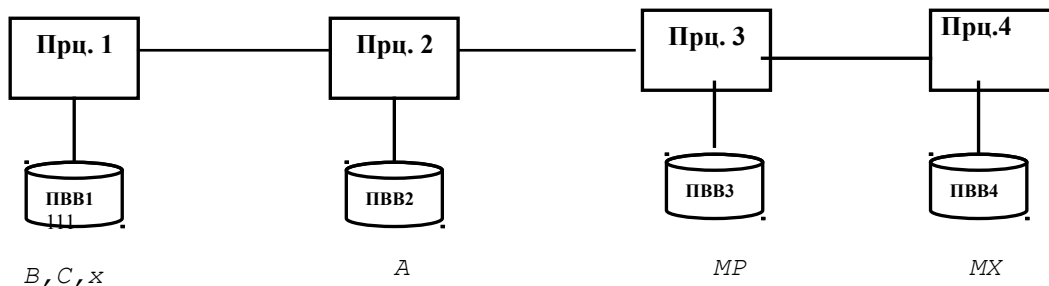


Рис. 2. Лінійна структура ПКС з локальною пам'яттю (ПКС ЛП) .

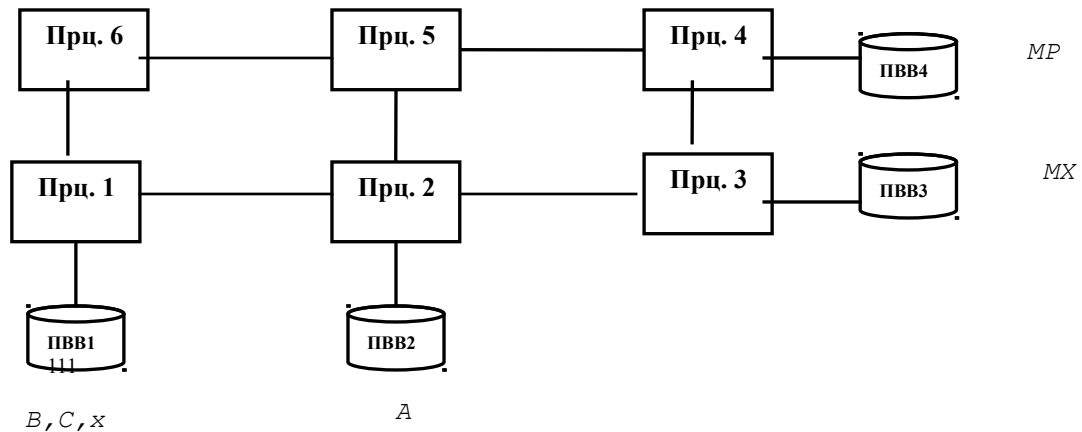


Рис. 3. Структура решітки для ПКС з локальною пам'яттю (ПКС ЛП)

Для лабораторних робіт 7-8 необхідне:

- розробити паралельний алгоритм рішення математичної задачі;
- описати алгоритм кожного процесу ( $T_1 - T_p$ );
- розробити структурну схему взаємодії задач (через механізм рандеву для лаб. 7 або через конструкції `send / receive MPI` для лаб 8);
- розробити програму
- виконати налагодження програми;
- отримати *правильні* результати обчислень.
- за допомогою Диспетчеру задач Windows проконтролювати завантаження ядер процесору.

## ЛАБОРАТОРНА РОБОТА 7

### Ада.Рандеву

**Мета роботи:** розробка програми для ПКС з ЛП

**Мова програмування:** Ада

**Засоби організації взаємодії процесів:** механізм рандеву

**Література:** [ 1, Розділ 6, с. 175-181.], [ 2, Цикл 7, с. 130 -134]

## ЛАБОРАТОРНА РОБОТА 8

### Бібліотека MPI

**Мета роботи:** розробка програми для ПКС з ЛП

**Мова програмування:** за вибором

**Засоби організації взаємодії процесів:** посилення повідомлень MPI (Send/Receive)

**Література:** [ 1, Розділ 6, с. 156-162.], [ 2, Цикл 7, с. 139 -146]

## ОФОРМЛЕННЯ ЗВІТУ З ЛАБОРАТОРНИХ РОБІТ

Звіт (протокол) повинен включати наступні розділи:

1. Титульний лист (номер роботи, тема роботи, ПІБ виконавця, групу)
2. Лист технічного завдання на роботу (ТЗ)  
(структура ПКС, математична задача, мова та засоби програмування)
3. Виконання роботи:
  - Етап 1. Розробка математичного **паралельного алгоритму** рушення задачі  
(наприклад,  $A_n = x * B_n + C_n$ , СР:  $x$ )
  - Етап 2. Розробка **алгоритму** для кожного процесу ( $T_1 - T_4$ )
  - Етап 3. Розробка структурної **схеми** взаємодії процесів (**Додаток А**)
  - Етап 4. Лістинг програми (з «шпакою»)

## ЛІТЕРАТУРА

### Основна

1. Жуков І., Корочкін О. Паралельні та розподілені обчислення – Київ: «Корнійчук», 2005.- 260 с.
2. Жуков І., Корочкін О. Паралельні та розподілені обчислення. Навч. посібн. 2-ге видання - Київ: «Корнійчук», 2014. - 284 с.
3. Zhukov I., Korochkin A. Parallel Computing. Kyiv: «Kornychuk», 2010.- 240 с.

### Додаткова

4. Антонов А.С. Параллельное программирование с использованием технологии OpenMP: Учебное пособие".-М.: Изд-во МГУ, 2009. - 77 с.
5. Богачев К.Ю. Основы параллельного программирования. – М.: БИНОМ. Лаб. знаний, 2003. – 342 с .
6. Брайант Р. Компьютерные системы: архитектура и программирование. - БНУ-СПб, 2005. - 1186 с.
7. Бройнль Т. Паралельне програмування. Початковий курс: Навч. посіб. – К.: Вища шк, 1997. – 358 с.
8. Валях Е. Последовательно-параллельные вычисления. – М.: Мир, 1985. – 456 с.
9. Воеводин В.В. Математические модели и методы в параллельных процессах. – М.: Наука, 1984. – 296 с.
10. Воеводин В., Воеводин В. Параллельные вычисления. БХВ- Петербург, 2002. 608 стр.
11. Гергель В. Высокопроизводительные вычисления для многопроцессорных многоядерных систем. . М.: Изд. МГУ.- 2010. – 544 с.
12. Гома Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений. – М.: ДМК Пресс, 2002. – 704 с.
13. Гофф Макс К. Сетевые распределенные вычисления. Достижения и проблемы. – М.: Кудиц-Образ, 2005. - 320 с.
14. Дейтел Д. Введение в операционные системы. – М.: Мир, 1989. – 360 с.
15. Джахани Н. Язык Ада. – М.: Мир, 1988. – 552 с.
16. Корнеев В.Д. Параллельное программирование в MPI. – Москва-Ижевск: "Институт компьютерных исследований", 2003. - 303 с.

17. Корочкин А.В. Ада95: Введение в программирование. – К.: Свит, 1999. – 260 с.
18. Линев А., Боголепов Д. Технологии параллельного программирования для процессоров новых архитектур. М.: Изд. МГУ.- 2010. – 160 с.
19. Лисков Б., Гатэг Дж. Использование абстракций и спецификаций при разработке программ : Пер. с англ. – М.: Мир, 1989. – 424 с.
20. Лупин С., Посыпкин М. Технологии параллельного программирования. М.: ИД Форум, 2011. - 208 с.
21. Мак-Дональд М., Шпуста М. Microsoft ASP.NET 2.0 с примерами на С# 2005 для профессионалов.: Пер. с англ. – М.: Изд. дом «Вильямс», 2006. - 1408 с.
22. Малышкин В.Э., Корнеев В.Д. Параллельное программирование мультимпьютеров. - НГТУ, 2006. - 296 с.
23. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: Общий подход. – М.: Лаборатория Базовых Знаний, 2004. – 406 с.
24. Миренков Н.Н. Параллельное программирование для многомодульных вычислительных систем. – М.: Радио и связь, 1989. – 320 с.
25. Немнюгин С., Стесик О. Параллельное программирование для многопроцессорных вычислительных систем. – СПб.: БХВ – Петербург, 2002. – 400 с.
26. Немнюгин С. А. Модели и средства программирования для многопроцессорных вычислительных систем. С.Петербургский ГУ, 2010. - 150 с.
27. Ноутон П., Шилдт Г. Java2: Пер. с англ. – СПб.: БХВ – Петербург, 2000. – 1072 с.
28. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем. – М.: Радио и связь, 1989, – 280 с.
29. Параллельные вычисления / Под ред. Г.Родрига – М.: Наука, 1986. – 376 с.
30. Пайл Я. Ада – язык встроенных систем. – М.; Финансы и статистика, 1984. –120 с.
31. Перминов О.Н. Введение в язык программирования Ада.– М.: Радио и связь, 1991 – 228 с.
32. Программирование на параллельных вычислительных системах/ Пер. с англ./ Р.Бэбб, Дж. Мак – Гроу и др.; под ред.Бэбба П. - М.: Мир, 1991. – 376 с.
33. Русанова О.В. Программное обеспечение компьютерных систем. Особенности программирования и компиляции. – К.: Корнійчук, 2003. – 94 с.
34. Симкин С., Барлетт Н., Лесли А. Программирование на Java. Путеводитель – К.: НИПФ “ДиаСофт Лтд.”, 1996. – 736 с.
35. Соловьев Г.Н., Никитин В.Д. Операционные системы ЭВМ. – М.: Высш. школа., 1989. – 255 с.
36. Стіренко С. Г., Грибенко Д. В., Зіненко А. І., Михайленко А. В. Засоби паралельного програмування. - К., 2011. - 181 с.
37. Траспьютеры. Архитектура и программное обеспечение: Пер. с англ./Под ред.Г.Харпа. – М.: Радио и связь, 1993. – 304 с.
38. Троелсон С. С# и платформа.NET. Библиотека программиста- СПб.: Питер, 2004. – 796 с.
39. Уильямс Э. Параллельное программирование на С++ в действии. ДМК, - 2012. - 672 с.
40. Хьюз К, Хьюз Т. Параллельное и распределенное программирование в С++. Пер. с англ.- М.: Радио и связь, 1986. – 240 с.
41. Хоар Ч. Взаимодействующие последовательные процессы. - М.: Мир, 1989. – 180 с.
42. Хокни Р., Джессхоул К. Параллельные ЭВМ. Пер. с англ.- М.: Радио и связь, 1986.–240 с.
43. Шилд Г. Полный справочник по С#.: Пер. с англ. – М.: Изд. дом «Вильямс», 2007.– 752 с.
44. Шамим Э., Джейсон Р. Многоядерное программирование Питер, 2010. - 180 с.
45. Эндрюс Г. Основы многопоточного, параллельного и распределенного программирования.: Пер. с англ. – М.: Изд. дом «Вильямс», 2003. – 512 с.
46. Breshears C. The Art of Concurrency. O'Really Media, 2009. – 280 p.
47. Burns A., Wellings A. Real-Time Systems and Programming Languages. Addison – Wesley, 2001, – 386 p.
48. Burns A., Programming in Occam2. Reading. Addison–Wesley, 1995, – 326 p.

49. Burns A., Wellings A. Concurrency in Ada. – Cambridge: Cambridge University Press, 1995., – 420 p.
50. El – Rewini H, Lewis T., Distributed and Parallel Computing.– Manning Pub. Co.1998, – 430 p.
51. Hyde P. Java Thread Programming. – Indianapolis, IN: Sams Publishing, 1999, – 324 p.
52. Hoare C.A.R. Monitors: An Operating System Structuring Concept, Communications of ACM, Vol.17, №.10, Oct.1974, pp. 549–557
53. Hoare C.A.R. Communicating Sequential Processes, Communications of ACM, vol.21, №. 8, Aug. 1978, pp. 666 – 667.
54. Hoare C.A.R. Communicating Sequential Processes,: Printice Hall, International Series in Computer Science, Englewood Cliffs NJ, 1985. – 186 p.
55. Goetz B. Java Concurrency in Practice.- Addison–Wesley Professional, 2006, – 384 p.
56. Korochkin A., Rusanova O. Scheduling Problems for Parallel and Distributed Systems– In Proceeding of the ACM Annual Conference (SIGADA'99) (The Redondo Beach, CA, USA, October 17–21, 2001) ACM Press, New York, NY, 1999, pp. 182– 190;
57. Korochkin A. Ada95 as a Foundation Language in Computer Engineering Education in Ukraine – In Proceeding of the Ada-Europe International Conference on Reliable Software Technologies (Ada-Europe'99), (Santander, Spain, June 7 – 11, 1999), Lecture Notes in Computer Science , – № 1622, Springer, 1999, pp. 62 – 70.
58. Korochkin A., Salah I, Korochkin D. Experimental Analyze Ada Program in Cluster System – In Proceeding of the ACM Annual Conference (SIGADA'05) (Atalanta, Georgia, USA, November 13–21, 2005) ACM Press, New York, NY, 2005, pp. 126 – 134.
59. Lea D. Concurrent Programming in Java: Design Principles and Patterns. Reading, MA: Addison – Wesley, 1999, – 344 p.
60. Mattson T., Sanders B., Massingill B. Patterns For Parallel Programming. Addison-Wesley, 2005. – 246 p.
61. Taft S., Duff R., Brukardt R., Ploedereder T. Consolidated Ada Reference Manual. Language and Standard Libraries, Springer, Berlin: 2001, – 562 p.
62. McKenney P. Is Parallel Programming Hard, And, If So, What Can You Do About It? [Електронний ресурс], <https://www.kernel.org/pub/linux/kernel/people/paulmck/perfbook/>
63. TOP500. [Електронний ресурс ]. [http://parallel.ru/computers/\\_top500.list42.html](http://parallel.ru/computers/_top500.list42.html)
64. Chapman B. Using OpenMP: portable shared memory parallel programming. Massachusetts Institute of Technology, London, 2008, - 350 p.
65. Intel Developer Zone [Електронний ресурс], <http://software.intel.com/ru-ru/articles/more-work-sharing-with-openmp>
66. Параллельные методы матричного умножения [Електронний ресурс], [www.hrcc.unn.ru/file.php?id=426](http://www.hrcc.unn.ru/file.php?id=426)
67. Офіційний сайт компанії Інтел. [Електронний ресурс], [www.intel.com](http://www.intel.com)
68. Форум MPI. [Електронний ресурс], <http://www.mpi-forum.org/docs/docs.html>
69. Who's Using Ada? Real-World Projects Powered by the Ada Programming Language. [Електронний ресурс], <http://www.seas.gwu.edu/~mfeldman/ada-project-summary.html>
70. The Special Interest Group on Ada (ACM's SIGAda). [Електронний ресурс], <http://www.sigada.org/>
71. Офіційний сайт організації Ada-Europe. [Електронний ресурс], <http://www.ada-europe.org/>
72. Офіційний сайт проекту GAP. [Електронний ресурс], <http://www.adacore.com/academia/universities/>
73. Офіційний сайт компанії AdaCore. [Електронний ресурс], <http://www.adacore.com>



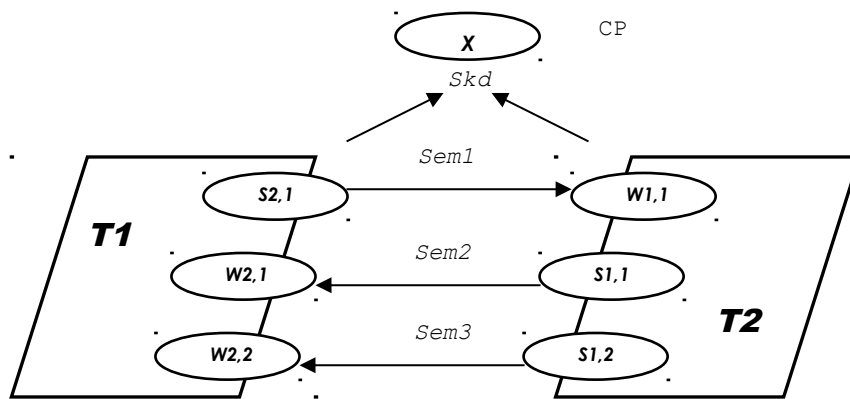
# Додатки

## Додаток А

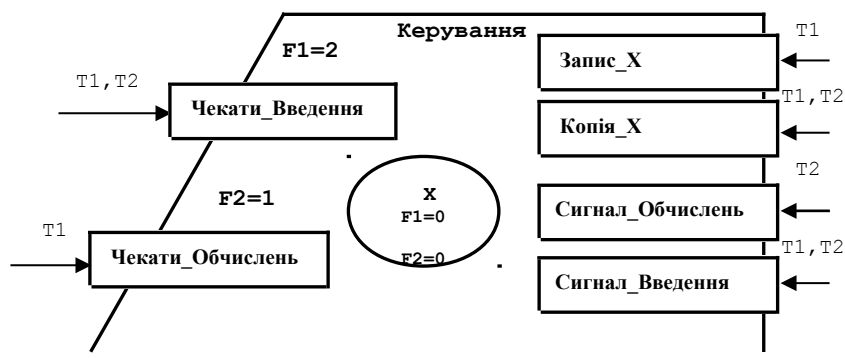
Приклади розробки схем взаємодії процесів

Для ПКС зі СП (цикл 1)

### Семафори

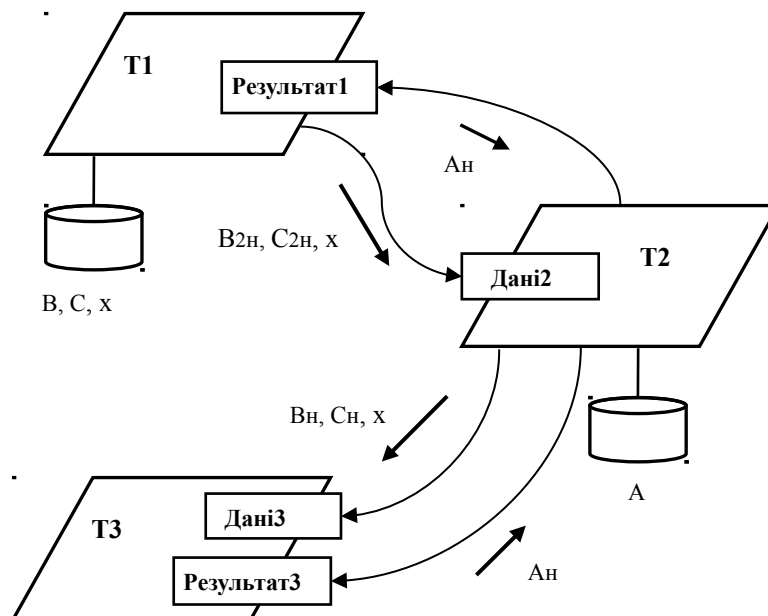


### Монітори (Захищений модуль)



### Для ПКС з ЛП (цикл 2)

Ада. Рандеву



### Умовні позначення:

$a$  – скаляр

$A$  – вектор розміру  $N$

$MA$  – матриця ( $N \times N$ )

$a = (B * C)$  - скалярний добуток векторів

$A = B * MC$  – множення вектора на матрицю

$MA = MB * MC$  – операція множення матриць

$a = \max(B)$  – операція пошуку максимального значення

$A = \text{sort}(D)$  – операція сортування вектору

## Додаток Б

### ВАРІАНТИ ЗАВДАНЬ

Варіанти завдань для лабораторних робіт надаються викладачем. (перший цикл для ПКС СП - роботи 1-6) та (другий цикл ПКС ЛП - роботи 7-8).

Студент може обрати від трьох до п'яти робіт в залежності від оцінки, яку він планує отримати.

	<b>Засоби програмування взаємодії потоків/ мови (бібліотеки) програмування</b>	<b>D/E 3 роботи</b>	<b>B/C 4 роботи</b>	<b>A 5 робіт</b>
1	Семафори, мютекси, події, критичні секції / Ada або Java або C# або WinAPI	1	2	2
2	Монітори/ Ada або Java	1	1	1
3	Бар'єри, критичні секції/ OpenMP	-	-	1
4	Посилання повідомлень/ Ada (рандеву) або MPI	1	1	1

## Додаток В

### Приклад виконання та оформлення лабораторної роботи

#### Лабораторна робота 1 Ада. Семафори

#### ЗАВДАННЯ НА РОБОТУ

##### 1. Структура паралельної комп'ютерної системи

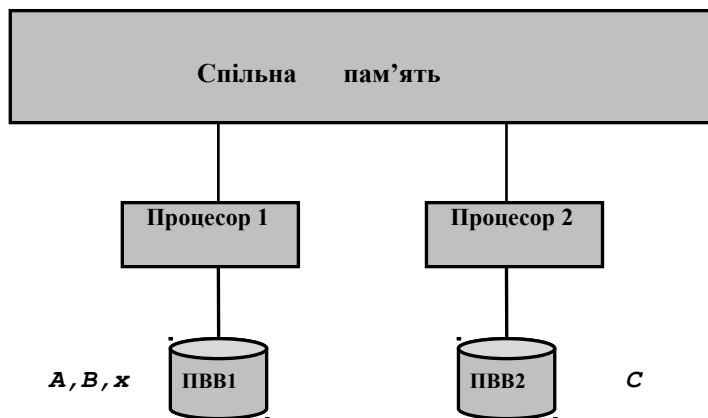


Рис. В.1. Структура ПКС СП

##### 2. Математична задача

$$A = B + C * x,$$

##### 2. Мова програмування: Ада

##### 3. Засоби програмування взаємодії потоків (задач) : семафори мови Ада (пакет `Ada.Synchronous_Task_Control`)

#### ВИКОНАННЯ РОБОТИ

##### Етап 1. Побудова паралельного алгоритму

$$A_n = B_n + C_n * x$$

Спільні ресурси:  $x$

##### Етап 2. Розроблення алгоритмів потоків (задач) .

Задача Т1

Точки

синхронізації

1. Введення  $B$  і  $x$
2. **Сигнал** задачі  $T2$  про введення  $B$  та  $x$  --  $S2-1$
3. **Чекати** на введення  $C$  у задачі  $T2$  --  $W2-1$
4. Копіювати  $x1 := x$  -- Критична ділянка (КД)
5. Обчислення  $A_n = B_n + C_n * x1$
6. **Чекати** на завершення обчислень в  $T2$  --  $W2-2$
7. Виведення результату  $A$

### Задача $T2$

1. Введення  $C$
2. **Сигнал** задачі  $T1$  про уведення  $C$  --  $S1-1$
3. **Чекати** на уведення даних в задачі  $T1$  --  $W1-1$
4. Копіювати  $x2 := x$  -- КД
5. Обчислення  $A_n = B_n + C_n * x2$
6. **Сигнал**  $T1$  про завершення обчислень --  $S1-2$

## Етап 3. Розроблення структурної схеми взаємодії задач.

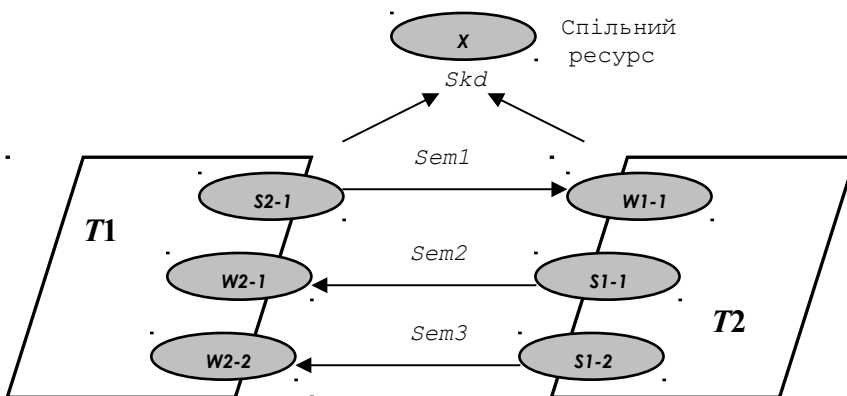


Рис 3.1 Структурна схема взаємодії задач

Призначення семафорів:

- **Skd** – для керування доступом до спільного ресурсу ( $x$ );
- **Sem1** – для синхронізації по завершенню введення в  $T1$ ;
- **Sem2** – для синхронізації по завершенню введення в  $T2$ ;
- **Sem3** – для синхронізації по завершенню обчислень в  $T2$  і виведенню результату.

## Етап 4. Розроблення програми

```

-----
-- Лаб. Робота №1
-- Ада.Використання семафорів
-- Операція  $A = B + C * x$ 
-- Сидоров В.І. гр. КС 421
-- 28.09.2019
-----

```

```
With Ada.Integer_Text_IO, Ada.Text_IO, Ada.Synchronous_Task_Control;
Use Ada.Integer_Text_IO, Ada.Text_IO, Ada.Synchronous_Task_Control;
```

```
procedure Lab1 is
```

```
  N: integer := 100; -- розмір векторів
  P: integer := 2;   -- кількість процесорів
  H: integer := N/P; -- розмір підвектора
```

```
  X: integer; -- спільний ресурс (глобальна змінна)  c
```

```
  type Вектор is array (1.. N) of integer;
```

```
  A,B,C : Вектор;          -- глобальні змінні
```

```
  -- семафори
  Sem1, Sem2, Sem3, Skd:  Suspension_Object;
```

```
  procedure Запуск_Задач is
```

```
    task T1;
```

```
    task body T1 is
```

```
      x1: integer;          -- локальна змінна
```

```
    begin
```

```
      put_line("Process T1 started ");
```

```
      -- введення B та x
```

```
      for i in 1.. N loop
```

```
        B(i) := 1;
```

```
      end loop;
```

```
      x := 2;
```

```
      -- сигнал задачі T2 про введення B та x
```

```
      Set_True(Sem1);
```

```
      -- чекати на введення даних в задачі T2
```

```
      Suspend_Until_True(Sem2);
```

```
      -- копіювати x в x1
```

```
      Suspend_Until_True(Skd);
```

```
      x1 := x;          -- критична ділянка
```

```
      Set_True(Skd);
```

```
      -- обчислення AH
```

```
      for i in 1 .. N loop
```

```
        A(i) := B(i) + C(i)*x1;
```

```
      end loop;
```

```
      -- Чекати на завершення обчислень AH в T2
```

```
      Suspend_Until_True(Sem3);
```

```
      -- виведення результату
```

```
      put_line(" A = ");
```

```
      for i in 1 .. N loop
```

```
        put(A(i),3);
```

```
      end loop;
```

```
      put_line("Process T1 finished");
```

```
    end T1;
```

```
    -----
```

```
    task T2;
```

```
    task body T2 is
```

```
      x2: integer;          -- локальна змінна
```

```
    begin
```

```
      put_line(" Process B started ");
```

```
      -- введення C
```

```
      for i in 1.. N loop
```

```
        C(i) := 2;
```

```
      end loop;
```

```

-- сигнал задачі T1 про уведення C
Set_True(Sem2);

-- чекати на уведення даних в задачі T1
Suspend_Until_True(Sem1);

--копіювати x в x2
Suspend_Until_True(Skd);
    x2:= x;          -- критична ділянка
Set_True(Skd);

-- обчислення АН
for i in N+1 .. N loop
    A(i):= B(i) + C(i)*x2
end loop;

-- сигнал T1 про завершення обчислень АН
Set_True(Sem3);

    put_line(" Process T2 finished");
end T2;

begin

    null;

    end Запуск_Задач;

-- тіло основної програми
begin

    put_line(" Main procedure started ");

    Set_True(Skd); -- встановлення початкового значення семафору Skd (true)

    Запуск_Задач;

end Lab1;

```

25 09 2019