Затверджено Вченою Радою ОНУ
імені І.І. Мечникова
від “20” грудня 2016 р. № 4

Одеський національний університет імені І.І. Мечникова
Кафедра математичного та комп’ютерного моделювання

“ЗАТВЕРДЖУЮ”
Проректор з науково-педагогічної роботи
_____
“____” _____ 20___ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК26 «Технології розподілених систем та паралельних обчислень»
(назва навчальної дисципліни)

Рівень вищої освіти _____бакалавр_____

Спеціальність _____126 – Інформаційні системи та технології_____
(код і назва спеціальності (тей)

Факультет _____математики, фізики та інформаційних технологій_____
(назва інституту, факультету)

2021

Робоча програма складена на основі навчальної програми з дисципліни «Технології розподілених систем та паралельних обчислень».

Розробник:
Вербицький В.В., к.ф.-м.н., доцент

Робоча програма затверджена на засіданні кафедри математичного та комп'ютерного моделювання

Протокол № 1 від. "30" 08 2021 р.

Завідувач кафедри _____ (В.В. Реут)
(підпис) (прізвище та ініціали)

Схвалено навчально-методичною комісією (НМК) з ІТ факультету МФІТ

Протокол № 1 від. "6" 09 2021 р.

Голова НМК _____ (О.С. Савастру)
(підпис) (прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____" _____ 20__ р.

Завідувач кафедри _____ (_____)
(підпис) (прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____" _____ 20__ р.

Завідувач кафедри _____ (_____)
(підпис) (прізвище та ініціали)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, спеціалізація, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	***заочна форма навчання***
Загальна кількість: кредитів – 5 годин – 150 залікових модулів – 2 змістових модулів – ІНДЗ* – РГР – 1 (вид завдання)	Галузь знань 12 Інформаційні технології (шифр і назва) Спеціальність 126 Інформаційні системи та технології (код і назва) Спеціалізації: - (назва) Рівень вищої освіти: бакалавр	Нормативна / за вибором (ВНЗ/студента)	
		Рік підготовки:	
		4-й	-й
		Семестр	
		7-й	-й
		Лекції	
		30 год.	год.
		Практичні, семінарські	
		0 год.	год.
		Лабораторні	
		30 год.	год.
		Самостійна робота	
		90 год.	год.
		Форма підсумкового контролю: екзамен,	

* – за наявності

2. Мета навчальної дисципліни

Дисципліна “Технології розподілених систем та паралельних обчислень” орієнтована на вивчення основних принципів побудови паралельних програм для паралельних та розподілених обчислювальних систем.

Мета курсу — отримання теоретичних знань та практичних навиків в використанні технологій написання паралельних та розподілених програм, що повинне забезпечити вміння створювати паралельні та розподілені програмні додатки для різноманітних комп’ютерних систем.

Для підготовки курсу використана велика низка новітніх наукових та навчально-методичних робіт як зарубіжних так і вітчизняних фахівців у галузі паралельного та розподіленого програмування, характеристики апаратного та програмного забезпечення комп’ютерів від провідних виробників, тощо.

Предметом дисципліни є технології побудови паралельних та розподілених програм.

Основними завданнями дисципліни є:

- отримання теоретичних знань і практичних навичок, достатніх для проектування та побудови паралельних додатків для сучасних комп’ютерів та розподілених обчислювальних систем;
- вивчення і реалізація основних алгоритмів, покладених в основу паралельного та розподіленого програмування.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей** (згідно ОПП «Інформаційні системи та технології»):

загальних:

- ЗК.01 Здатність до дослідницької роботи;
- ЗК.02 Здатність до роботи в команді;
- ЗК.03 Здатність аналізувати та синтезувати науково-технічну, природничо-наукову та загальнонаукову інформацію,

фахових:

- ФК.01 Здатність до математичного та логічного мислення, знання основних понять, положень і методів фундаментальної математики та вміння їх використовувати під час розв’язання конкретних виробничих завдань;
- ФК.04 Знання сучасних методів побудови та аналізу ефективних алгоритмів і вміння їх реалізувати в конкретних застосуваннях;
- ФК.07 Знання теоретичних особливостей чисельних методів, можливостей їх адаптації до інженерних задач, уміння використовувати чисельні методи під час розв’язання різнопланових прикладних задач;
- ФК.11 Знання сучасних технологій та інструментальних засобів розробки програмних систем, уміння їх застосовувати на всіх етапах життєвого циклу;
- ФК.15 Знання архітектури та стандартів компонентних моделей, комунікаційних засобів і розподілених обчислень, уміння розв’язувати проблеми масштабованості, підтримки віддалених компонентів і взаємодії різних програмних платформ в розподілених корпоративних інформаційних системах рівня підприємства;
- ФК.24 Знання основ архітектури комп’ютерів і комп’ютерних мереж, уміння застосовувати їх в процесі обґрунтування технічного забезпечення ІС;
- ФК.28 Ґрунтовна математична підготовка, а також підготовка з теоретичних, методичних і алгоритмічних основ інформаційних технологій для використання математичного апарату під час вирішення прикладних і наукових завдань в області інформаційних систем і технологій;
- ФК.29 Ґрунтовна підготовка в області програмування, володіння алгоритмічним мисленням, методами програмної інженерії для розробки програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик;

Програмні результати навчання:

- ПРН 01. Уміння здійснювати аналіз інформаційних технологій та предметних областей створюваних інформаційних систем, обробляти результати аналізу, осмислювати та подавати їх, обґрунтовувати запропоновані рішення на сучасному науково-технічному рівні.
- ПРН 04. Уміння опановувати та розробляти документацію на системи, продукти і сервіси в сфері інформаційних технологій, а також спілкуватися державною та іноземною мовами.
- ПРН 06. Уміння використовувати основні поняття, ідеї та застосовувати методи фундаментальної та прикладної математики під час розв'язання конкретних задач в сфері інформаційних технологій.
- ПРН 07. Уміння застосовувати стандарти, профілі, специфікації інформаційних технологій, що визначають функціональні можливості, динаміку поведінки, протоколи взаємодії та інші характеристики систем, продуктів і сервісів інформаційних технологій під час їх розробки та впровадження.
- ПРН 10. Уміння розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання професійних завдань в області комп'ютерних наук, а також виконувати їхню програмну реалізацію у вигляді системного та прикладного програмного забезпечення інформаційних систем і технологій
- ПРН 11. Уміння використовувати апаратні засоби сучасних систем обробки інформації, обчислювальних систем та мереж різного призначення
- ПРН 13. Уміння застосовувати парадигми та мови програмування, мови опису інформаційних ресурсів, мови специфікацій, а також інструментальні засоби проектування і створення систем, продуктів і сервісів інформаційних технологій
- ПРН 15. Уміння працювати в різних операційних системах та розробляти елементи відповідного системного програмного забезпечення
- ПРН 18. Уміння застосовувати методи та алгоритми комп'ютерної графіки у процесі розробки графічних застосувань, проектувати та створювати системи мультимедіа і графічного моделювання
- ПРН 24. Уміння враховувати конфігурацію і особливості комп'ютерів та сучасних мікроконтролерів при розробці інформаційних систем, а також програмувати їх та з'єднувати в єдиний комплекс з використанням каналів передачі даних
- ПРН 25. Уміння експлуатувати глобальні, локальні, мобільні та інші комп'ютерні мережі та будувати на їх основі розподілені інформаційні системи
- ПРН 27. Вміти складати тести та контрольні приклади для оцінки якості програмних систем та виявлення помилок їхнього функціонування.

Очікувані результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати:

1. основні парадигми паралельного та розподіленого програмування;
2. основні прийоми написання паралельних програм з використанням потоків стандарту POSIX;
3. технологію OpenMP написання паралельних програмних додатків для SMP-систем;
4. технологію MPI написання розподілених паралельних програм для розподілених комп'ютерних систем та мереж,

вміти:

1. створювати паралельні алгоритми для розв'язування різноманітних задач;
2. аналізувати ефективність паралельного алгоритму;
3. реалізувати паралельний алгоритм для копьютерних систем з різними паралельними архітектурами;

Місце навчальної дисципліни в структурі освітнього процесу.

Вивчення дисципліни безпосередньо базується на знаннях і вміннях, отриманих студентами при опануванні дисциплін „Введення в сучасні операційні системи і середовища“, „Програмування“, „Структури даних та алгоритми“ .

Отримані знання й навички студенти можуть використати при написанні випускних і дипломних робіт.

Курс складається з наступних частин: теоретичної (лекції), практичної (завдання, лабораторні роботи, тести для самоперевірки) та самостійної роботи.

Лабораторні роботи виконуються в операційній системі Linux з використанням компілятора GCC, який підтримує відкритий стандарт для написання паралельних програм OpenMP та дозволяє використовувати реалізацію Open MPI стандарту MPI для написання паралельних додатків для розподілених систем. Реалізація Open MPI вільно розповсюджується.

3. Програма навчальної дисципліни

Модуль 1

Тема 1. Вступ до предмету. Паралелізм. Два основних підходи до досягнення паралелізму: паралельне програмування, розподілене програмування. Найпростіша модель паралельного програмування PRAM. Найпростіша модель розподіленого програмування. Процес проектування паралельних і розподілених програм: декомпозиція, зв'язок, синхронізація. Базові рівні програмного паралелізму. Середовища для паралельного і розподіленого програмування.

Тема 2. Класифікація паралельних обчислювальних систем: SMP-системи, кластери, MMP-системи. Системи із загальною й розподіленою пам'яттю. Архітектура NUMA та ccNUMA. Організація когерентності багаторівневої ієрархічної пам'яті в SMP - системах.

Тема 3. Головні парадигми паралельного програмування. Ітеративний паралелізм (множення матриць). Рекурсивний паралелізм(адаптивна квадратура). «Виробники та споживачи» (конвейери). канали ОС Unix. Передача повідомлень: «керівник -робітник», конвеєр. Проблеми паралельного і розподіленого програмування: «гонка» даних, нескінченна відстрочка, взаємоблокіровка, труднощі організації зв'язку. Типові задачі синхронізації паралельних процесів: задача взаємного виключення, «виробник-споживач», «читачі-письменники», « філософи, що обідають».

Тема 4. Паралельне програмування: потоки стандарту POSIX. Створення потоку в Unix. Атрибути потоку. Очікувані й від’єднанні потоки. Скасування потоку: асинхронне скасовування, синхронне скасовування, потоки, які не можна скасувати. Потокові дані. Оброблювач очищення. Очищення потокових даних у C++. Потокові семафори. Мютекси. Умовні змінні POSIX. Приклад використання мютексів і потокових семафорів.

Модуль 2

Тема 5. Технологія паралельного програмування OpenMP. Призначення OpenMP. Модель OpenMP-додатка. Директива паралельної обрабки parallel. Директива розподілення роботи for. Директиви розподілення роботи sections та section. Директиви single та master. Директиви tasks та taskwait. Директиви синхронізації barrier, ordered, critical, atomic. Спільні та приватні змінні. Функції середовища виконання. Функції блокування та синхронізації. Змінні оточення. Алгоритми планування паралельного виконання циклів (static, dynamic, guided, runtime scheduling).

Тема 6. Технологія розподіленого програмування MPI. Призначення MPI. Модель MPI-додатка. Комунікатори. Функції ініціалізації й завершення роботи. Етапи передачі повідомлень між паралельними процесами MPI. Типи даних MPI. Функції передачі повідомлень між процесами типу «один-одному». Колективні комунікації. Розподілені операції в MPI. Створення нових типів даних MPI. Створення розподілених операцій. Топології процесів. Створення

декартової топології процесів в MPI-додатках. Приклад використання декартової топології процесів.

Тема 7. Моделювання та аналіз паралельних обчислень. Модель обчислень у вигляді графа "операції-операнди". Граф інформаційних залежностей послідовного алгоритму. Опис схеми паралельного виконання алгоритму. Характеристики часу виконання паралельного алгоритму. Теореми про оцінки часу виконання паралельного алгоритму. Показники ефективності паралельного алгоритму. Каскадна схема сумування. Модифікована каскадна схема сумування. Оцінка максимально допустимого паралелізму (закон Амдаля, ефект Амдаля, закон Густавсона – Барсиса). Аналіз масштабованості паралельних обчислень.

4. Структура навчальної дисципліни

Т Е М А	Кількість годин									
	Денна форма					Заочна форма				
	Всього	У тому числі				Всього	У тому числі			
		л	лаб	інд	сам		л	лаб	інд	сам
1	2	3	4	5	6	7	8	9	10	11
Модуль 1										
Тема 1. Вступ до предмету.	6	2			4					
Тема 2. Класифікація паралельних обчислювальних систем:	8	2			6					
Тема 3. Головні парадигми паралельного програмування.	14	4			10					
Тема 4. Паралельне програмування: потоки стандарту POSIX	36	6	10		20					
Тема 5. Технологія паралельного програмування OpenMP.	36	6	10		20					
Тема 6. Технологія розподіленого програмування MPI	36	6	10		20					
Тема 7. Моделювання та аналіз параллельних обчислень.	14	4			10					
Усього за курс	150	30	30		90					

5. Теми семінарських занять

Не передбачено

6. Теми практичних занять

Не передбачено

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
Модуль 1		
1	Створення та очікування завершення потоку. Очікувані та від'єднані потоки	2
2	Мютекси та потокові семафори	4
3	Умовні (сигнальні) змінні	4
Модуль 2		

4	Технологія OpenMP. Директиви OpenMP.	4
5	Технологія OpenMP. Загальні та приватні змінні	4
6	Технологія OpenMP. Потокобезпечні функції	2
7	Технологія MPI. Обмін повідомленнями між двома процесами	4
8	Технологія MPI. Створення типів даних і глобальних операцій	4
9	Топологія процесів MPI	2

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
	Модуль 1	
1	Вступ до предмету.	4
2	Класифікація паралельних обчислювальних систем:	6
3	Головні парадигми паралельного програмування.	10
4	Паралельне програмування: потоки стандарту POSIX	20
	Модуль 2	
4	Технологія паралельного програмування OpenMP.	20
5	Технологія розподіленого програмування MPI	20
6	Моделювання та аналіз параллельних обчислень.	10

9. Індивідуальні завдання

Варіант завдання .

Математичне сподівання є популярним показником мінливості або різноманітності даних, що використовуються в статистиці. Якщо у нас є певний набір даних, кожна точка даних є раціональним або раціональним числом (наприклад, висота людини, виражена в см.), то мале математичне очікування вказує на те, що точки даних мають тенденцію бути дуже близькими до середнього значення, тоді як велике відхилення вказує, що точки даних розподілені по великому діапазону значень, далеко від середнього значення.

Для вектора $X = \{x_1, x_2, \dots, x_n\}$ його середнє значення $mean(x) = \frac{x_1+x_2+\cdots+x_n}{n}$.

Тоді математичне очікування X є величина

$$\sigma_X = \sqrt{\frac{\left(x_1 - mean(X)\right)^2 + \left(x_2 - mean(X)\right)^2 + \cdots + \left(x_n - mean(X)\right)^2}{n}}$$

Припустимо тепер, що у нас є два набори даних рівного розміру, вектор X і Y і ми хотіли б з'ясувати, чи існує певна кореляція між значеннями xi і yi. Наприклад, якщо xi і yi представляють собою висоту і вагу людини, відповідно, то, можливо, існує певна кореляція.

Існують різні способи вимірювання кореляції між двома наборами даних. Мабуть, найпопулярнішим є коефіцієнт кореляції Пірсона, який визначається наступним чином. Для двох наборів даних рівного розміру, які визначаються векторами X і Y, коефіцієнт кореляції Пірсона визначається так

$$\rho_{X,Y} = \frac{\frac{1}{n}\sum_{i=1}^{i=n}\left(x_i - mean(X)\right)\cdot\left(y_i - mean(Y)\right)}{\sigma_X \cdot \sigma_Y}.$$

Коефіцієнт кореляції Пірсона завжди має значення між -1 і 1. Чим ближче до 1, тим більше показник кореляції між значеннями даних.

Необхідно написати програму, яка обчислює коефіцієнт кореляції Пірсона, використовуючи MPI. Вхідні дані задаються в двох масивах a і b значень типу double. Довжина кожного масиву складає 1000000. Потрібно написати програму на мові C, яка:

1. реалізує послідовний алгоритм (не використовуючи паралелізм) для обчислення коефіцієнта кореляції Пірсона даних, представленого в масивах a і b; реалізує паралельний алгоритм з використанням MPI для обчислення того ж самого;

2. для заданих даних обчислюється кореляцію Пірсона за послідовним і паралельним алгоритмами;

3. роздруковує результати обох алгоритмів, час, що витрачається обома алгоритмами, і прискорення, досягнуте шляхом розпаралелювання;

Швидкість роботи програм оцінюватися з використанням конкретної ініціалізації масивів: a [i] = sin (i) і b [i] = sin (i + 1).

Вам необхідно надати:

1. вихідний код програми на C;

2. короткий звіт з аналізом прискорення вашої програми, досягнутого шляхом розпаралелювання

10. Методи навчання

Підчас проведення лекцій використовуються пасивний та активний методи навчання. Під час виконання студентами лабораторних робіт використовується активний методи навчання.

11. Методи контролю

На лабораторних заняттях проводиться усне опитування студентів. За результатами виконання лабораторних робіт студенти оформляють звіти. Звіт перевіряється викладачем в присутності студента в формі співбесіди.

Екзамен (7-й семестр).

12. Питання для підсумкового контролю

Модуль 1

1. Два основних підходи до досягнення паралелізму: паралельне програмування, розподілене програмування.
2. Найпростіша модель паралельного програмування PRAM.
3. Найпростіша модель розподіленого програмування.
4. Процес проектування паралельних і розподілених програм: декомпозиція, зв'язок, синхронізація.
5. Базові рівні програмного паралелізму.
6. Середовища для паралельного і розподіленого програмування.
7. SMP-системи, кластери, MMP-системи.
8. Системи із загальною й розподіленою пам'яттю.
9. Архітектура NUMA та ccNUMA.
10. Організація когерентності багаторівневої ієрархічної пам'яті в SMP - системах.
11. Ітеративний паралелізм (множення матриць).
12. Рекурсивний паралелізм(адаптивна квадратура). «Виробники та споживачі» (конвейери).
13. Проблеми паралельного і розподіленого програмування: «гонка» даних, нескінченна відстрочка, взаємоблокіровка, труднощі організації зв'язку.
14. Типові задачі синхронізації паралельних процесів: задача взаємного виключення, «виробник-споживач», «читачі-письменники», « філософи, що обідають».
15. Створення потоку в Unix. Атрибути потоку. Очікувані й від’єднанні потоки.
16. Скасування потоку: асинхронне скасовування, синхронне скасовування, потоки, які не можна скасувати.
17. Потокові дані. Оброблювач очищення. Очищення потокових даних у C++.
18. Потокові семафори. Мютекси.
19. Умовні змінні POSIX. Приклад використання мютексів і потокових семафорів.

Модуль 2

1. Призначення OpenMP. Модель OpenMP-додатка.
2. Директива паралельної обрабки parallel.
3. Директива розподілення роботи for.

4. Директиви розподілення роботи sections та section.
5. Директиви single та master.
6. Директиви tasks та taskwait.
7. Директиви синхронізації barrier, ordered, critical, atomic.
8. Спільні та приватні змінні.
9. Функції середовища виконання.
10. Функції блокування та синхронізації.
11. Змінні оточення OpenMP.
12. Алгоритми планування паралельного виконання циклів (static, dynamic, guided, runtime scheduling).
13. Призначення MPI. Модель MPI-додатка.
14. Комунікатори. Функції ініціалізації й завершення роботи.
15. Етапи передачі повідомлень між паралельними процесами MPI.
16. Типи даних MPI.
17. Функції передачі повідомлень між процесами типу «один-одному».
18. Колективні комунікації.
19. Розподілені операції в MPI.
20. Створення нових типів даних MPI.
21. Створення розподілених операцій.
22. Топології процесів.
23. Створення декартової топології процесів в MPI-додатках.
24. Приклад використання декартової топології процесів.
25. Модель обчислень у вигляді графа "операції-операнди".
26. Граф інформаційних залежностей послідовного алгоритму.
27. Опис схеми паралельного виконання алгоритму.
28. Характеристики часу виконання паралельного алгоритму.
29. Теореми про оцінки часу виконання паралельного алгоритму. Показники ефективності паралельного алгоритму.
30. Каскадна схема сумування.
31. Модифікована каскадна схема сумування.
32. Оцінка максимально допустимого паралелізму (закон Амдаля, ефект Амдаля, закон Густавсона – Барсиса).
33. Аналіз масштабованості паралельних обчислень.

12. Розподіл балів, які отримують студенти
Залік за 7-й семестр

Поточне тестування та самостійна робота								**Підсумковий тест (екзамен)**	Сума
Модуль 1				Модуль 2					
Т1	Т2	Т3	Т4	Т5	Т6	Т7		25	100
3	12	12	15	15	15	15			

Т1, Т2 ... Т9 – теми змістових модулів.

Шкала оцінювання ECTS та національна

Сума балів за всі види навчальної діяльності	Оцінка за шкалою ECTS	Оцінка за національною шкалою	
		Для екзамену, курсового проекту (роботи), практики	Для заліку
90 - 100	A	відмінно	Зараховано
75-89	BC	добре	
60-74	DE	задовільно	

35-59	FX	Незадовільно з можливістю повторного складання	Не зараховано з можливістю повторного складання
1-34	F	Незадовільно з обов'язковим повторним вивченням дисципліни	Не зараховано з обов'язковим повторним вивченням дисципліни

13 Методичне забезпечення

1. Методическое пособие и варианты контрольных заданий по курсу «Параллельное программирование». Часть I. Сост. Вербицкий В.В. – Одесса, «Укрполиграф», 2006 – 34 с.
2. Методическое пособие и варианты контрольных заданий по курсу «Параллельное программирование». Часть II. Сост. Вербицкий В.В. – (доступно через Internet по адресу www.odu.enu.ua)

14. Рекомендована література

Базова

1. Robert Robey, Yuliana Zamora. Parallel and High Performance Computing. – Manning Publications Co., 2021. – 705 p.
2. R. Trobec, B. Slivnik, P. Bulid, B. Robik. Introduction to Parallel Computing. – Springer International Publishing, 2017. – 380 p.
3. Грегори Р. Эндрюс. Основы многопоточного, параллельного и распределенного программирования. – М.: Издательский дом «Вильямс», 2003. – 512 с.
4. Гергель В.П. Теория и практика параллельных вычислений. М.: Национальный Открытый Университет "ИНТУИТ", 2016. — 501 с.
5. Качко Е. Г. Параллельное программирование: Учебное пособие. — Харьков: "Форт", 2011. — 528 с.
6. Богачёв К.Ю. Основы параллельного программирования. – М.: Бином, 2003. – 342с.

Допоміжна

1. Митчелл М., Оулдем Дж., Самьюэл А. Программирование для Linux. Профессиональный подход. – М: Издательский дом "Вильямс", 2002.
2. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб: БХВ-Петербург, 2002.
3. Воеводин В.В. Математические основы параллельных вычислений. М: МГУ, 1991.
4. Корнеев В.В. Параллельные вычислительные системы. – М: Ноллидж, 1999.
5. Олифер В.Г., Олифер Н.А. Сетевые операционные системы. – СПб: Питер, 2001.
6. Стивенс У. UNIX: взаимодействие процессов. – СПб: Питер, 2003. – 576 с.
7. Ian Foster. Desinging and building parallel program. – Addison-Westly, 1995. (доступна через Internet по адресу www.parallel.ru)
8. Mare Shir, Steve Otto. MPI: The complete reference. – MIT Press, 1996. (доступна через Internet по адресу www.parallel.ru)

15. Інформаційні ресурси

1. http://www.mcs.anl.gov – Argonne National Laboratory, Center for Computational Science and Technology.
2. http://www.cacr.caltech.edu/ – CACR - Центр Вычислительных Исследований (Center of Advanced Computing Research)
3. http://netlib.org – Netlib is a collection of mathematical software, papers, and databases.