

ПРОЄКТУВАННЯ ВІЗУАЛЬНОГО СЕРЕДОВИЩА ДЛЯ НАВЧАННЯ СТУДЕНТІВ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ

Сіциліцин Ю. О.

старший викладач кафедри інформатики і кібернетики

Мелітопольський державний педагогічний університет імені Богдана Хмельницького

вул. Гетьманська, 20, Мелітополь, Запорізька область, Україна

orcid.org/0000-0002-3888-5575

yurarud@gmail.com

Ключові слова: паралельні обчислення, паралельне програмування, C++, MPI, навчання паралельного програмування.

У статті висвітлений підхід до навчання студентів паралельного програмування засобами візуального навчального середовища для розробки паралельних програм. Автор висвітлює проблеми, які виникають у навчанні студентів паралельного програмування на лабораторних роботах. Натепер у закладах вищої освіти не починають викладати паралельне програмування, поки студент не розвине знання з послідовного програмування. Така практика приводить до того, що у розробці паралельних програм студенти спочатку пишуть послідовну програму. Автор доводить, що розробка студентами паралельних програм на лабораторних роботах з паралельного програмування потребує досить великого рівня володіння мовою програмування та прийомами роботи з даними. Це пов'язане з використанням динамічних масивів мови програмування C++ та бібліотеки паралельного та розподіленого програмування MPI. Для переходу до розробки паралельного коду спочатку студенту потрібно розробити частину програми, пов'язану з оголошенням змінних і масивів та опису роботи з динамічними масивами. Також розробка паралельних алгоритмів ускладнюється тим, що студенти вже встигли виробити у себе навички створення послідовних алгоритмів, а ці навички заважають розробці паралельних алгоритмів. Для вирішення проблеми автор пропонує розробити навчальне візуальне середовище. Для цього автор формулює основні вимоги до проєктування візуального середовища та розробляє формальну мову, яка буде використовуватися паралельним середовищем. Ця мова допоможе спростити етап розробки програми, пов'язаний з підготовкою даних до паралельної обробки, та зосередитися саме на паралельній обробці. У висновках автор стверджує, що навчальне візуальне середовище допоможе студентам розробити паралельний код та візуально відстежити його виконання, що своєю чергою допоможе студентам ефективніше засвоїти паралельне програмування.

DESIGNING VISUAL ENVIRONMENT FOR TRAINING STUDENTS PARALLEL PROGRAMMING

Sitsylitsyn Yu. O.

*Senior Lecturer at the Department of Computer Science and Cybernetics
Bogdan Khmelnsky Melitopol State Pedagogical University
Hetmanska str., 20, Melitopol, Zaporizhzhia region, Ukraine
orcid.org/0000-0002-3888-5575
yurarud@gmail.com*

Key words: *parallel calculations, parallel programming, C ++, MPI, teaching students parallel programming.*

The article highlights the approach to teaching students parallel programming by means of a visual learning environment for the development of parallel programs. The author highlights the problems that arise when teaching students parallel programming in laboratory work. Today, higher education institutions do not begin to teach parallel programming until the student develops knowledge of sequential programming. This practice leads to the fact that when developing parallel programs, students first write a sequential program. The author proves that the development of parallel programs by students in laboratory work on parallel programming requires a sufficiently high level of proficiency in the programming language and methods of working with data. This is due to the use of dynamic arrays of the C ++ programming language and the library of parallel and distributed MPI programming. To proceed with the development of parallel code, the student must first develop a part of the program related to the declaration of variables and arrays and a description of working with dynamic arrays. Also, the development of parallel algorithms is complicated by the fact that students have already developed the skills to create sequential algorithms, and these skills hinder the development of parallel algorithms. To solve the problem, the author proposes to develop a visual learning environment. To do this, the author formulates the basic requirements for the design of the visual environment and develops a formal language that will be used in a parallel environment. This language will help to simplify the stage of program development related to the preparation of data for parallel processing and focus on parallel installments. In conclusion, the author argues that the learning visual environment will help students develop parallel code and visually track its implementation, which in turn will help students learn parallel programming more effectively.

Постановка проблеми. У закладах вищої освіти зі спеціальності «Комп'ютерні науки» студентів навчають послідовного програмування за допомогою однопоточної моделі програмування. Оскільки апаратне забезпечення комп'ютера стає все більш паралельним, виникає більша потреба в інженерах програмного забезпечення, які мають досвід у розробці паралельних програм не лише шляхом «паралелізації» послідовних конструкцій. Навчання студентів паралелізму на початкових курсах зі спеціальності «Комп'ютерні науки» є важливим кроком до формування компетентностей майбутніх інженерів-програмістів.

Натепер у закладах вищої освіти не починають викладати паралельне програмування, поки студент не розвине знання з послідовного програ-

мування. Така практика приводить до того, що у розробці паралельних програм студенти спочатку пишуть послідовну програму [3]. Під час написання такої програми вони використовують уже вивчені послідовні алгоритми, а потім переробляють послідовну програму на паралельну. Такий підхід розробки паралельних програм приводить до виникнення у паралельних програмах типових помилок, таких як «стан гонитви» або «взаємне блокування».

Вирішення завдань на послідовній та паралельній обчислювальній системі значно відрізняються за кількістю етапів та їх змісту [2]. Тому для проєктування паралельних або розподілених програм потрібно вивчати паралельні та розподілені алгоритми обробки даних.

Великий вплив чинить на структуру і технологію проектування паралельних програм состав методів паралельної обробки даних: рівень обчислювальних архітектур, рівень чисельних методів, рівень технологій програмування, рівень мов програмування, рівень паралельної обробки.

Деякі дослідники для формування у студентів основ паралельного мислення пропонують почати вивчення паралельного програмування вже з першого курсу [4].

Але на першому курсі більшість студентів ще не володіє достатньою мірою або зовсім не володіє основами мов програмування та середовищ розробки, які потрібні для початку вивчення паралельного програмування. Тому ми слідом за дослідниками [6] сформулюємо основні вимоги до проектування візуального середовища для навчання студентів паралельних та розподілених обчислень.

Мета статті – сформулювати основні вимоги до проектування візуального середовища для навчання студентів паралельних обчислень.

Фундаментальні дослідження у галузі підготовки майбутніх інженерів-програмістів у закладах вищій освіти здійснюють В. Биков, В. Осадчий, З. Сейдаметова, О. Спірін, Ю. Триус. Також науковці висвітлюють окремі аспекти зазначених проблем, зокрема: питання якості професійної підготовки програмістів (А. Власюк, В. Круглик, К. Осадча, А. Стрюк); вимоги до професійних якостей майбутніх інженерів-програмістів (Н. Длугунівич, А. J. Ко, P. L. Li, J. Zhu); організацію навчання майбутніх інженерів-програмістів у вищих інших країн світу (В. Круглик). Огляд та вибір засобів паралельного програмування зробили С. Лупін, М. Посипкін, В. Воеводін, Е. Уільямс та інші вчені. І. Скопін досліджує навчання паралельного програмування, науковець А. Шеметова робить огляд методичних прийомів навчання паралельного програмування.

Спочатку визначимось з платформою, для якої потрібно проектувати візуальне середовище. Студенти будуть використовувати візуальне середовище протягом лабораторних робіт з дисципліни «Паралельне програмування». Тому можливі два варіанти: середовище буде розташоване на сервері або середовище буде розташоване локально на персональному комп'ютері у класі. Основною операційною системою сучасних серверів висту-

пає Linux. На персональному комп'ютері операційна система – це Windows або Linux. Тому візуальне середовище потрібно розробляти кросплатформним, тобто воно повинне однаково працювати як під операційною системою Linux, так і під операційною системою Windows.

Серед кросплатформних мов, які можна застосувати для розробки візуального середовища, можна виділити мови Java та Python. Але якщо використовувати візуальне середовище через браузер як додаток до навчальної платформи, то можна зазначити, що деякі із сучасних браузерів (наприклад Google Chrome) не підтримують мову програмування Java. Тому мовою програмування візуального середовища вибираємо мову програмування Python.

Візуальне середовище у навчанні студентів паралельних обчислень повинне вирішувати дві проблеми цього навчання.

Перша проблема – для розробки паралельних програм потрібно на високому рівні знати розробку послідовних програм. Натепер одним з основних інструментів розробки паралельних програм є бібліотека MPI [8]. Ця бібліотека працює у взаємодії з мовою програмування C++ або Фортран. Розробка паралельної програми мовою програмування C++ з бібліотекою MPI потребує від студентів дуже хорошого знання мови програмування C++, а саме розділів про динамічну роботу з пам'яттю, використання вхідних та вихідних потоків даних, циклів тощо. У разі написання паралельної програми засобами C++ програми студент багато часу та уваги приділяє саме написанню коду, налагодженню роботи програми та усуненню помилок. Цей процес значною мірою відволікає увагу студента від розробки правильного паралельного коду та в умовах обмеження часу в рамках лабораторної роботи приводить до написання неправильного паралельного коду або до неможливості написання програми загалом.

Друга проблема – довга розробка послідовних програм заважає розвитку «паралельного мислення». Під «паралельним мисленням» будемо мати на увазі такий спосіб алгоритмічного мислення, який дозволяє побудувати паралельний алгоритм. Структура паралельного стилю мислення може бути представлена у вигляді спрощеної моделі (рис. 1).

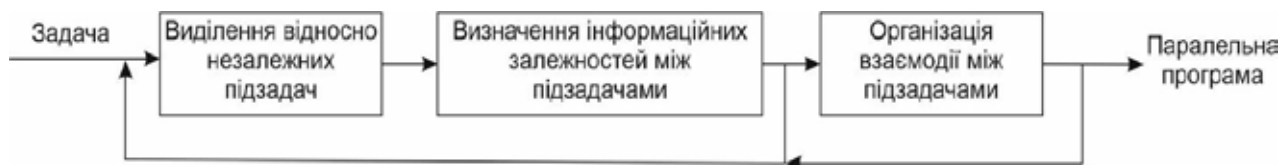


Рис. 1. Паралельний стиль мислення

Основні етапи розробки паралельного алгоритму можна охарактеризувати таким чином: виділення підзадач, встановлення інформаційних залежностей і організація інформаційної взаємодії підзадач. Крім того, важливу роль відіграє розуміння того, що всі підзадачі, насправді, виконуються паралельно (одночасно). Як показує практика, цей факт декларується студентам на словах, але фактично не сприймається і не розуміється повною мірою.

Виходячи з описаних вище проблем, можна сформулювати основні вимоги до візуального середовища, яке буде займатися навчанням студентів розробки програм для паралельних обчислень.

Для вирішення першої проблеми у візуальному середовищі, яке розробляється, розподілимо оператори мови C++ на блоки. У розробці паралельної програми за допомогою візуального середовища студент не повинен багато часу приділяти синтаксису та пунктуації мови програмування, тому розробимо синтаксичні конструкції у вигляді блоків. Кожний з таких блоків повинен відповідати за закінчену смислову частку коду. Крім того, необхідно забезпечити логічну послідовність та відповідність блоків конструкціям мови програмування C++ та бібліотеки MPI.

Розглянемо основні конструкції мови C++, які необхідні для розробки паралельних програм:

- оголошення змінних;
- оператори введення-виведення на консоль (printf, scanf);
- оператори роботи з файлами (fstream);
- оголошення одновимірних масивів;
- оголошення багатовимірних масивів;
- заповнення масивів випадковими числами;
- виділення частки масиву з повного масиву (по строках, по стовпчиках, довільно). Зведемо ці дані у таблицю да дамо назви операторам (табл. 1).

Нами були розглянуті конструкції для роботи з масивами, які є основними складниками процесу паралельного програмування.

Якщо подивитись на бібліотеку MPI [8], то можна зробити висновок, що всі оператори бібліотеки вже повністю оптимізовані і не потребують конструювання з них блоків. Для спрощення роботи з ними потрібно убрати позначки роботи з динамічною пам'яттю.

Перейдемо до другої проблеми. Для її вирішення у середовищі, яке проектується, потрібно розробити графічний аналог кожного з блоку операторів та візуально відображати зв'язки по

Таблиця 1

Оператори візуального середовища для навчання студентів паралельних обчислень

Опис	Оператор	Приклад на мові C++
Оголошення змінної	int (double) <i>змінна</i>	int a
Введення з консолі	input (<i>змінна</i>)	int a; scanf ("%d", &a);
Виведення на консоль	output (<i>змінна</i>)	int a; printf ("%d", a);
Читання з файлу	input_file (<i>ім'я файлу</i>)	#include <fstream> ifstream fin ("1.doc"); if (fin.is_open()) fin.getline(buff, 50); fin.close();
Виведення у файл	output_file (<i>ім'я файлу</i>)	#include <fstream> ofstream fout ("1.txt"); fout << "текст"; fout.close();
Оголошення одновимірних масивів	array (<i>розмір, змінна</i>)	int n; cin >> n; double* a = new double[n];
Оголошення багатовимірних масивів	array (<i>рядок, стовпчик, змінна</i>)	int n; int m; cin >> n >> m; double** a = new double*[n]; for (int i = 0; i < n; i++) {a[i] = new double[m];}
Заповнення масивів випадковими числами	fill_array (<i>змінна</i>)	srand(time(0)); for (int i = 0; i < n; i++) {a[i] = rand() % 10 + 1}
Виділення частки масиву з повного масиву	slice_array (<i>початок, кількість, крок, змінна</i>)	Приклади можна побачити тут [1]

даних та по командах між цими частинами. Таким чином, студент під час розробки паралельної програми буде одразу бачити потік даних між частинами коду.

При цьому будуть відображені основні проблеми паралельного програмування, які виникають під час розробки програми:

- взаємні блокування паралельних ділянок;
- несинхронний доступ або гонки;
- нескінченна відстрочка;
- взаємоблокування.

Висновки. Розробка візуального навчального середовища для навчання студентів паралельних обчислень завдяки виключенню з процесу розробки паралельної програми фактора помилок «звичайного» програмування та додавання візуального середовища відстеження потоку даних та команд під час виконання паралельної програми дозволить зосередити увагу студентів саме на паралельному програмуванні, чим підніме якість вивчення дисципліни «Паралельні обчислення».

ЛІТЕРАТУРА

1. Грицюк Ю.І., Рак Т.Є. Програмування мовою C++ : навчальний посібник. Львів : Вид-во Львівського ДУ БЖД, 2011. 292 с.
2. Карпов А. Введение в проблематику разработки параллельных программ. URL: <https://www.viva64.com/ru/a/0016/>.
3. Adams J., Brown R., Shoop E. Patterns and Exemplars: Compelling Strategies for Teaching Parallel and Distributed Computing to CS Undergraduates. J. *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*.
4. Breuer A., Bader M. Teaching Parallel Programming Models on a Shallow-Water Code. *11th International Symposium on Parallel and Distributed Computing*. 2012.
5. Franczak T., Nkansahz A., Marrinan T., Papka M.E. A Path from Serial Execution to Hybrid Parallelization for Learning HPC. *Workshop on Education for High-Performance Computing ser. EduHPC'17*. 2017.
6. Gregg C., Tychonievich L., Cohoon J., Hazelwood K., EcoSim J. *A language and experience teaching parallel programming in elementary school, in SIGCSE'12*. 2012, pp. 51–56.
7. Kuhail M.A., Cook S., Neustrom J.W., Rao P. Teaching Parallel Programming with Active Learning. *IEEE International Parallel and Distributed Processing Symposium Workshops*. 2018.
8. Sitsylitsyn Y. Methods, tools for teaching parallel, distributed computing in universities: a systematic review of the literature. *ICHTML 2020: SHS Web of Conferences 75*. 2020.

REFERENCES

1. Gritsyuk, Y.I., Cancer, T.E. (2011). Prohramuvannya movoyu C++ : navchal'nyy posibnyk [C++ programming : a textbook]. Lviv : Lviv State University BJD Publishing House. 292 p. [in Ukrainian].
2. Karpov, A. Vvedeniye v problematiku razrabotki parallel'nykh program [Introduction to the development of parallel programs]. URL: <https://www.viva64.com/ru/a/0016/> [in Russian].
3. Adams, J., Brown, R., Shoop, E. (2013). Patterns and Exemplars: Compelling Strategies for Teaching Parallel and Distributed Computing to CS Undergraduates . *IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*.
4. Breuer, A., Bader, M. (2012). Teaching Parallel Programming Models on a Shallow-Water Code. *11th International Symposium on Parallel and Distributed Computing*.
5. Franczak, T., Nkansahz, A., Marrinan, T., Papka, M.E. (2017). A Path from Serial Execution to Hybrid Parallelization for Learning HPC. *Workshop on Education for High-Performance Computing ser. EduHPC'17*.
6. Gregg C., Tychonievich L., Cohoon J., Hazelwood K., EcoSim J. (2012). A language and experience teaching parallel programming in elementary school, in *SIGCSE'12*, 2012, pp. 51–56.
7. Kuhail. M.A., Cook S., Neustrom J.W., Rao P. (2018). Teaching Parallel Programming with Active Learning. *IPDPSW*.
8. Sitsylitsyn, Y. (2020). Methods, tools for teaching parallel, distributed computing in universities: a systematic review of the literature. *ICHTML 2020: SHS Web of Conferences*.