****

*BT- Form 1*

University Unit (Faculty):

**Faculty of Computer Science and Engineering\_(CSE)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Application for Topic and Mentor of Bachelor Thesis**

**Applicant’s Data**

Name and Surname: Astrit Zendeli

Index no.: 669

Study Program: 4 Year Study Program

Contact info: Tel. 071 620 834E-mail: astrit.zendeli@cse.uist.edu.mk

**Bachelor Thesis Title**

“Speedup of Advanced Programs in C language with Parallel Programming”

**Explanation of the Scientific, Artistic and Professional Viability of the Topic**

* **Topic of Research**

The Research topic is mainly related to Parallel Programming or Computing, which is a type of computation where the calculation or execution of programs, processes, or tasks is achieved simultaneously. We can divide programs into several parts, each part executing at the same time (simultaneously).

Using Parallel Programming or Computing we can speed-up the execution of different programs so that we can save time and resources. Most programs that are more complicated to calculate take longer to finish, this is where parallel programming helps us to complete these complicated programs in a faster and efficient way.

The topic of research is based mainly on the part of speeding-up advanced C programs in Multi-Core systems.

* **Topic Justification**

One of the essential characteristics of a programmer is solving problems in a faster and efficient way, but sometimes to do this parallel execution is needed, because it gives better results than sequential execution.

The main purpose of choosing this topic is to observe and research if the speed of advanced programs coded in the C language can be increased using this method or not.

Choosing and working on this topic will help me in my career as a computer science student to find alternative and better solutions in solving everyday problems in a much more effective, efficient, and quicker way.

Nearly all the giant IT companies use distributed systems that require the processing of tasks in either Multi-Processor or Multi-Core level. Getting a job in this field requires parallel programming skills and experience. This is why the chosen research topic will help me in accomplishing this kind of a goal of being a greater programmer, problem solver, and increasing my chances of getting a better job.

Besides this, in the general case this type of programming has a lot of potential to help other people in different ways and benefit them. One example is that we live in a world where each day a lot of data is collected (Big Data) and processing this data to get specific results is needed. This represents a complex problem.

By using parallel programming concepts programmers, analysts, researchers, scientists, statisticians etc. can process this data faster and obtain the desired results quickly. So people can analyze traffic, weather, medical, satellite and many other types of data in a faster way and by using less resources (computational power). This allows finding solutions quickly and helping people that are in need (dependent) of those solutions.

A similar example is writing programs for clients that solve their specific problems, and of course solving them faster is very important for them. Here are included network, database, customer management problems and many others.

This is one of the main advantages of parallel approach, that’s why this topic is chosen to prove that solving complex problems will be faster with Parallel Computation and that it will be useful for everyone.

* **Research Methodology**

The first step is to gather all the required information regarding the topic of relevant books, research articles, and significant online resources. Establishing the working environment is crucial, this includes choosing the right software programs to work with.

After that, choosing the right problems that need to be solved and trying to solve them in parallel is the core part of this project that requires proper analyzing skills. When the right problems are found, proper algorithms will be written to solve them and different techniques will be used to find better solutions and optimize the code by implementing data structures, advanced algorithms, writing cleaner code, getting better results etc.

The most important methodology that will be used will be based on the measurement of the execution time of the sequential and parallel codes tested for different CPU’s. The execution times will be compared and using specific formulas the overall program speedup will be calculated (Amdahl’s Law).

The obtained test result will determine the conclusion of the topic.

* **Research Goals and Objectives**
* Finding better solutions to advanced or complex problems and making them execute faster using parallel programming methods.
* Finding out if a problem or program is faster executed sequentially or in parallel and depending on the situation concluding whether is it good to use a parallel approach or not.
* Deciding when it is good to use parallel code and in which situations it is appropriate.
* Learning the proper ways of implementing a parallel approach for Multi-Core based systems.
* Finding optimal solutions to specific problems.
* Becoming better in this field by adding more to my knowledge and improving my problem-solving skills.
* Increasing research capabilities and becoming a better programmer.
* **Expected Outcomes**

By the end of this thesis, it is expected that all the chosen problems are to be solved (in the C language) using parallel execution methods and to have the wanted results, which means that the sequential code needs to be slower than the parallel code in the given program with the condition that both of the codes give the correct answers or outputs.

It is expected that with proper analyses and research to conclude that in which cases we can solve a problem in a parallel manner and to analyze if the sequential approach would be appropriate for some situations because not all tasks can be solved in parallel. The main outcome that matters is getting faster solutions to given tasks.

* **Structure of the Bachelor Thesis**

In the first part of the thesis, an introduction will be given about the main software and technologies that will be used for parallel programming (OpenMP), then a brief explanation will be given about the advanced or complex problems that will be chosen to be solved in parallel (Ex. Knapsack, N-queens, Matrix Multiplication etc.). It is considered that three problems are to be chosen mainly and maybe more if there is enough time.

In the second part, each chosen problem will be explained, it will provide sequential and parallel code with an explanation of how the codes work. After that, statistics (in the form of tables) with execution times will be provided so that we can compare the obtained results. Each of the programs (solutions) will be tested on different CPU’s (laptops or desktop computers).

In the third part, it will be discussed in general whether or not a parallel approach is appropriate for any situation and how to distinguish when to execute the code sequentially or in parallel.

In the fourth and final part, the overall conclusion that we derive from the whole work (research) done will be discussed and presented.

**References**

[1] Pacheco, P. (2011). *An Introduction to Parallel Programming*(1st edition). Morgan Kauffman. [Online]. Available:

<https://b-ok.cc/book/998262/c3b0f5> [Accessed: 28-May-2020]

[2] Stallings, W. (2010). *Computer Organization and Architecture: Designing for Performance* (8th edition). Prentice Hall. [Online]. Available:

<https://www.inspirit.net.in/books/academic/Computer%20Organisation%20and%20Architecture%208e%20by%20William%20Stallings.pdf> [Accessed: 28-May-2020]

[3] Chandra, R., Dagum, L., Kohr, D., Maydan, D., McDonald, J., Menon, R. (2000). *Parallel Programming in OpenMP* (1st edition). Morgan Kaufmann. [Online]. Available:

<https://b-ok.cc/book/654975/d15e53> [Accessed: 28-May-2020]

[4] Quinn, M. J. (2003). *Parallel Programming in C with MPI and OpenMP*. McGraw Hill. [Online]. Available:

<https://b-ok.cc/book/1219129/53f53a> [Accessed: 28-May-2020]

[5] Kesri, V. & V., Pattnaik, P.K (2012).An Unique Solution for N Queen Problem, *in* *International Journal of Computer Applications (0975 – 8887) Volume 43* (No.12).[Online]. Available:

<https://www.researchgate.net/publication/258651417_An_Unique_Solution_for_N_queen_Problem> [Accessed: 28-May-2020]

DOI: 10.5120/6152-8539

[6] Crawford, M., &Toth,D.(2013).Parallelization of the Knapsack Problem as an Introductory Experience in Parallel Computing*, in Journal of Computational Science Education, Volume 4* (Issue 1). [Online]. Available:

<https://pdfs.semanticscholar.org/63ae/393c4e971ff4df71a485ad0221e02e0cb072.pdf> [Accessed: 28-May-2020]

[7] Bhugul, A.M (2017).Parallel computing using OpenMP, *in International Journal of Computer Science and Mobile Computing, Volume 6* (Issue.2), 90-94. [Online]. Available:

<https://www.ijcsmc.com/docs/papers/February2017/V6I2201713.pdf> [Accessed: 28-May-2020]

[8] Open Coding Community that Helps Developers Answer their Questions [Online]. Available: <https://stackoverflow.com/>

[9] Online Learning Platform for Students/Developers [Online]. Available:

<https://www.geeksforgeeks.org/>

[10] Online Learning Platform for Students/Developers [Online]. Available:

<https://www.tutorialspoint.com/>

**Proposals for Mentor**(\*to name 3 (three) mentors by student’s priority)

1. Dijana Capeska Bogatinoska (Associate Professor,PhD)

**Date** **Applicant**

Astrit Zendeli

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(signature)