|  |  |  |
| --- | --- | --- |
| **Identification** | Title  | Parallel Programming |
| Department | Computer Science & Engineering |
| Program | M.Sc. - 4 credits |
| Semester | Fall 2017 |
| Instructor | Seyed Amir Hossein Siahpooshha (PhD) |
| E-mail: | siahpooshha@gmail.com |
| Classroom/hours | Neftchilar Campus, room# |
| Pre-requisites | B.Sc. Computer Science & Engineering |
| Language | English (intermediate level) |
| Type | Major (Compulsory) |
| **Course Resources** | 1. Designing and Building Parallel Programs, by Ian Foster(Concepts and Tools for Parallel Software Engineering)<http://www.mcs.anl.gov/~itf/dbpp/text/book.html>Front Cover2. Class notes |
| **Course Objective** | This course provides a thorough discussion of parallel algorithm design, performance analysis, and program construction, illustrating fundamental principles with numerous examples. It approaches parallel programming as an engineering activity, in which programs are developed in a methodical fashion and both cost and performance are considered at each stage in a design. It is intended as both an introduction to parallel programming and a practitioner's guide for programmers, engineers, and scientists developing programs for parallel and distributed computer systems. |
| **Evaluation** | Midterm Exam | 40% |
| Final Exam | 60% |
| **Weekly Lectures** |
| 1 | [Overview](https://computing.llnl.gov/tutorials/parallel_comp/#ModelsOverview)* [What is Parallel Computing?](https://computing.llnl.gov/tutorials/parallel_comp/#Whatis)
* [Why Use Parallel Computing?](https://computing.llnl.gov/tutorials/parallel_comp/#WhyUse)
* [Who is Using Parallel Computing?](https://computing.llnl.gov/tutorials/parallel_comp/#Who)
 |
| 2 | [Concepts and Terminology](https://computing.llnl.gov/tutorials/parallel_comp/#Concepts)* [von Neumann Computer Architecture](https://computing.llnl.gov/tutorials/parallel_comp/#Neumann)
* [Flynn's Classical Taxonomy](https://computing.llnl.gov/tutorials/parallel_comp/#Flynn)
 |
| 3 | * [Some General Parallel Terminology](https://computing.llnl.gov/tutorials/parallel_comp/#Terminology)
* [Limits and Costs of Parallel Programming](https://computing.llnl.gov/tutorials/parallel_comp/#LimitsCosts)
 |
| 4 | [Parallel Computer Memory Architectures](https://computing.llnl.gov/tutorials/parallel_comp/#MemoryArch)* [Shared Memory](https://computing.llnl.gov/tutorials/parallel_comp/#SharedMemory)
* [Distributed Memory](https://computing.llnl.gov/tutorials/parallel_comp/#DistributedMemory)
 |
| 5 | * [Hybrid Distributed-Shared Memory](https://computing.llnl.gov/tutorials/parallel_comp/#HybridMemory)

[Parallel Programming Models](https://computing.llnl.gov/tutorials/parallel_comp/#Models)* [Overview](https://computing.llnl.gov/tutorials/parallel_comp/#ModelsOverview)
 |
| 6 | * [Shared Memory Model](https://computing.llnl.gov/tutorials/parallel_comp/#ModelsShared)
* [Threads Model](https://computing.llnl.gov/tutorials/parallel_comp/#ModelsThreads)
 |
| 7 | * [Distributed Memory / Message Passing Model](https://computing.llnl.gov/tutorials/parallel_comp/#ModelsMessage)
* [Data Parallel Model](https://computing.llnl.gov/tutorials/parallel_comp/#ModelsData)
 |
| 8 | Midterm Exam |
| 9 | * [Hybrid Model](https://computing.llnl.gov/tutorials/parallel_comp/#Hybrid)
* [SPMD and MPMP](https://computing.llnl.gov/tutorials/parallel_comp/#SPMD-MPMD)
 |
| 10 | [Designing Parallel Programs](https://computing.llnl.gov/tutorials/parallel_comp/#Designing)* [Automatic vs. Manual Parallelization](https://computing.llnl.gov/tutorials/parallel_comp/#DesignAutomatic)
* [Understand the Problem and the Program](https://computing.llnl.gov/tutorials/parallel_comp/#DesignUnderstand)
 |
| 11 | * [Partitioning](https://computing.llnl.gov/tutorials/parallel_comp/#DesignPartitioning)
* [Communications](https://computing.llnl.gov/tutorials/parallel_comp/#DesignCommunications)
* [Synchronization](https://computing.llnl.gov/tutorials/parallel_comp/#DesignSynchronization)
 |
| 12 | * [Data Dependencies](https://computing.llnl.gov/tutorials/parallel_comp/#DesignDependencies)
* [Load Balancing](https://computing.llnl.gov/tutorials/parallel_comp/#DesignLoadBalance)
* [Granularity](https://computing.llnl.gov/tutorials/parallel_comp/#DesignGranularity)
 |
| 13 | * [I/O](https://computing.llnl.gov/tutorials/parallel_comp/#DesignIO)
* [Debugging](https://computing.llnl.gov/tutorials/parallel_comp/#DesignDebug)
* [Performance Analysis and Tuning](https://computing.llnl.gov/tutorials/parallel_comp/#DesignPerformance)
 |
| 14 | [Parallel Examples](https://computing.llnl.gov/tutorials/parallel_comp/#Examples)* [Array Processing](https://computing.llnl.gov/tutorials/parallel_comp/#ExamplesArray)
* [PI Calculation](https://computing.llnl.gov/tutorials/parallel_comp/#ExamplesPI)
 |
| 15 | * [Simple Heat Equation](https://computing.llnl.gov/tutorials/parallel_comp/#ExamplesHeat)
* [1-D Wave Equation](https://computing.llnl.gov/tutorials/parallel_comp/#ExamplesWave)
 |
| 16 | Final Exam - |