Course #:	**B424/B524**
Course Title:	**Parallel and Distributed Programming**
Course Type:	Upper level elective
Prerequisites:	C243 Data Structures, M301 Linear Algebra.
Credits:	3
Text Book:	B. Wilkinson, M. Allen (1999): *Parallel Programming*, Prentice-Hall.
References:	• Handouts • G. R. Andrews (2000): *Foundations of Multithreaded, Parallel and Distributed Programming*, Addison-Wesley. • W. Gropp, E. Lusk, A. Skjellum (1999): *Using MPI*, second edition, The MIT Press.
Current Catalog Description:	Overview of parallel computers, shared memory, message passing, MIMD and SIMD classifications. Understanding and use of message passing and synchronization facilities such as MPI. Study of parallel programming models such as master-slave, client-server, tast-farming, divide-and-conquer and pipelining. Performance analysis of parallel systems, execution time, time complexity, load balancing and scalability.
Course Goals	The student who completes this course: 1. Will be introduced to the major aspects of parallel programming, both in a distributed memory model and in a shared memory model. 2. Will be able to identify and understand the basic functions of Parallel Computing. 3. Will be introduced to parallel computing architectures. 4. Will understand parallel software implementation and debugging issues. 5. Will develop an awareness of and appreciation for hardware/operating system/parallel program interaction. 6. Will be able to explore multiprocessing software models and tools.
Major Topics Covered in the Course	1. Classification and history of parallel computers 2. Sharing resources, synchronization 3. Master-slave model 4. Deadlocks, debugging 5. Evaluating parallel programs 6. Distributed memory models 7. Introduction to MPI 8. Divide and conquer 9. Computing with possible interruption 10. Pipeline models, interactive peers

<table>
<tr><td></td><td>11. Operating systems simulations
12. Client-server
13. Producers and consumers
14. Massively parallel algorithms
15. Introduction to pthreads
16. Critical region, semaphores</td></tr>
<tr><td>Laboratory projects (specify number of weeks on each)</td><td>Three Laboratories, half a week each (1.5h)
Lab 1: Introduction to MPI, compiling and executing a small parallel program

Lab 2: Introduction to the Pthread library, compiling and executing a small program using pthreads.

Lab 3: Working on a program showcasing the use of monitors in the context of the Pthread library.</td></tr>
<tr><td>Estimate Curriculum Category Content (Semester hours)</td><td><table>
<tr><td>Area</td><td>Core</td><td>Advanced</td></tr>
<tr><td>Algorithms</td><td>10</td><td>20</td></tr>
<tr><td>Software Design</td><td>4</td><td></td></tr>
<tr><td>Comp. Arch.</td><td></td><td>4</td></tr>
<tr><td>Data Structures</td><td>4</td><td></td></tr>
<tr><td>Prog. Languages</td><td></td><td>5</td></tr>
</table>
Additional hours may be dedicated to curriculum categories not listed above. For example explanation of concepts and theories. Discussion of social and ethical issues, discussion of inter personal relationships and working within groups.</td></tr>
<tr><td>Oral and Written Communications</td><td>Every student is required to submit at least __1 to 3___ written reports (not including exams, tests, quizzes, or commented programs) of typically ___2 to15__ pages and to make ___1__ oral presentations of typically __15___ minute’s duration.</td></tr>
<tr><td>Social and Ethical Issues</td><td>Not a course objective.</td></tr>
<tr><td>Theoretical Content</td><td>• Classification and history of parallel computers, 2h
• Theoretical models of parallel computing, 2h</td></tr>
<tr><td>Problem Analysis</td><td>A major project must be designed and implemented by each student.</td></tr>
<tr><td>Solution Design</td><td>Students present one or more solution designs for each problem and discuss the differences between them.</td></tr>
<tr><td>Prepared By</td><td>Vrajitoru, Wolfer</td></tr>
</table>