COURSE DESCRIPTION

Department and Course Number	CS 370	**Course Coordinator**	Raj Pamula
Course Title	Parallel and Distributed Programming	**Total Credits**	4

Current Catalog Description:

Parallel programming techniques; abstract models of hardware and operating systems to support parallel programs.

Textbook:

None

References:

Instructor notes on the faculty web page.

Course Goals:

At the end of the course, students have a good understanding of

- Parallel hardware architectures
- Parallel programming models
- Parallel algorithms
- Writing parallel programs

These course goals contribute to the success of **Student Learning Outcomes 1.a, 1.b, 1.d, 1.f, 2, 3, 5, and 6**.

Prerequisites by Topic:

- Data Structures
- Algorithms
- High level programming languages
- Operating Systems
- Computer Networks

Major Topics Covered in the Course:

- Parallel hardware architectures
- Parallel programming models
- Shared memory model
    1. Operating system support
    2. Shared memory access
    3. Parallel programming
- Distributed memory model

1. Clusters
2. Operating system support
3. Message Passing Interface
4. Parallel programs

- Parallel Algorithms
- Efficiency and timing analysis

Laboratory Projects (specify number of weeks on each):

The students complete 5-6 lab projects on a selected topic, (1-2 weeks per project)

- Shared memory programming : Using “forking and joining” principle to do a computation
- Shared memory programming : Using “shared memory” principle to do a computation
- Shared memory programming : Using “barrier” principle to do a computation
- Distributed memory programming : Using MPI library to do a computation
- A large project (such as sorting and searching) on Shared and Distributed memory platforms.

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms		.75	Data Structures		.75
Software Design		1.5	Prog. Languages		0.5
Comp. Arch.		0.5			

Oral and Written Communications:

The students are required to submit the source code and software documentation of the programming project.

Social and Ethical Issues:

No significant component.

Theoretical Content:

- Introduction to parallel architectures (1 week)
- Introduction to parallel programming modes (1 week)
- Introduction to parallel algorithms (1 week)

Problem Analysis:

Students write parallel programs using a particular parallel architecture.

Solution Design:

Solution design involves parallel programs on a given architecture using Operating Systems support and other libraries. Students execute programs using both “Shared Memory” and “Distributed Memory” models.