Course-Plan

School **Engineering**
Department **Computer Science and Engineering**
Course Code **CS 502**
Course Name **System Software**
Instructor **Utpal Sharma**

1. **Abstract**: CS 502 System Software is a course that covers the working principles, design and implementation of software such as assemblers, macro-processors, linkers, loaders, debuggers, text editors, user interfaces, etc. These are general purpose software which make the computer usable for solving real problems of end-users in various ways. The course covers theory as well as programming practice in the laboratory.

2. **Objective:**

Module	Topic	Learning Objectives
1	Introduction	To understand what is System Software and what are the different types
2	Featues of Assembly languages	To understand the features of Assembly languages so as to understand what an assembler has to do.
3.	Assembler Design and Implementation	To learn how to design an assembler, including details of essential data structures, and implementation through lab practice.
4.	Macro-processors	To understand what are macros, macro-processors, and design of a macro-processor for an assembly language.
5.	Program linking and loading	To understand what is program linking and loading, the various types of linking, and the design of linkers and loaders. Also the students would be familiarised with these features in Linux and MS-Windows.
6.	Debuggers and Text Editors	To understand the role of important software tools debuggers and text editors and the challenges in design of these.
7.	Software tools	To familiarise the students with various software tools (other than debuggers and text editors).
8.	Compiler	To understand the basic functions and structure of compilers.

3. Prerequisites of the course:
Fair knowledge of computer architechture and organization, C programming, and data structures would be required.

4. Course outline (See Syllabus)

Suggested reading:

Text Book:

Dhamdhere, D M : Systems Programming and Operating Systems, Tata Mc Graw Hill

Reference Books:

- **Sumitabha Das** : Unix System V.4 Concepts and Applications, Tata Mc Graw Hill
- Linux Manuals.
- Windows Manuals.

5. (a)**Time-Plan**

Tentative Lectures	Topics
1	Overview of system software, types of system software, language procesors
2-5	Features of assembly languages, the task of an assembler Assembly process - pass structure of assembler, assembler data structures Forward referencing, location counter Exercise: Progamming exercise in C for file operations and distinction between text file and binary file.
6-8	Lexical Analysis- implementation methods, *flex* interface between lexical analyser and the rest an assembler Exercise: Implement a lexical analyser for a particular assembly language using using string-comparision method, and then by using *flex*.
9-10	Strucure and use of Mnemonics table Analysis phase, structure and usage of symbol table Exercise: Define mnemonic table and symbol table in C.
11-12	Intermediate code- possible formats Synthesis phase - write m/c code into output file Exercise: Implement an assembler for a simple assembly language. Familiarisation with an assembler in Windows/Linux
13-15	Macros and macro pre-processing - parameter passsing methods for macros, data structures for macro expansion. Exercise: Implement a simple macro-preprocessor for an assembly language
16-17	Program linking and loading, need for relocation Structures of object modules Program linking : Static linking and Dynamic linking Exercise: See the structure of an object module.
18-20	Case studies of linking schemes in Linux and Windows, shared objects and DLLs

21-22	OLE and ActiveX features of MS Windows
23-24	Debuggers: Features, implementation Example of debugger - *gdb*
25-26	Text editors: types, features, implementation Case studies - *vi, sed, MS-Word* Exercise: Implement a simple text editor.
27-28	Software tools: *make, grep, awk, rcs, sed, sort, join* Exercise: Exercise with these tools. *Shell* programming.
29-31	Compilers: Features of HLLs, compiler's task, grammars, stages of compilation, optimization

Term Assignment:

Each student shall have to implement one of the following-

- a simple assembler
- a simple macro-preprocessor
- a simple linker
- a simple text editor

They shall have to create the program, prepare a write-up and submit these by end of Octo 2012. The evaluation will include questions related to the work.

(b) Evaluation plan

Term Assignments	15
Class Assignments	10
Mid Term	60
Lab Assignments	25
Attendence and discipline	5
End-term lab test	25
End Term	100
Total	**240**

6. Pedagogy :

Teaching-learning methods to be used

Lecture and Discussion
Term assignment
Class assignments

7. Expected outcome: Towards the end of the course the student would understand common system software so as to be able to use available such software comfortably as well as create required such software or improve existing ones as required in actual professional work.