GUJARAT TECHNOLOGICAL UNIVERSITY

SYSTEM PROGRAMMING

SUBJECT CODE: 2150708
B.E. 5th SEMESTER

Type of course: System Programming

Prerequisite: Data Structures and Operating Systems

Rationale: NA

Teaching and Examination Scheme:

<table>
<thead>
<tr>
<th>Teaching Scheme</th>
<th>Credits</th>
<th>Examination Marks</th>
<th>Total Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L T P C</td>
<td>Theory Marks</td>
<td>Practical Marks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ESE (E)</td>
<td>PA (M)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 0 2 6</td>
<td></td>
<td>70</td>
<td>20</td>
</tr>
</tbody>
</table>

Content:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Content</th>
<th>Total Hrs</th>
<th>% Weightage</th>
</tr>
</thead>
</table>
| 1       | Overview of System Software  
Introduction, Software, Software Hierarchy, Systems Programming,  
Machine Structure, Interfaces, Address Space, Computer Languages,  
Tools, Life Cycle of a Source Program, Different Views on the Meaning  
of a Program, System Software Development, Recent Trends in Software  
Development, Levels of System Software | 06 | 10% |
| 2       | Overview of Language Processors  
Programming Languages and Language Processors, Language  
Processing Activities, Program Execution, Fundamental of Language  
Processing, Symbol Tables  
Data Structures for Language Processing: Search Data structures,  
Allocation Data Structures. | 06 | 15% |
| 3       | Assemblers  
Elements of Assembly Language Programming, Design of the  
Assembler, Assembler Design Criteria, Types of Assemblers, Two-Pass  
Assemblers, One-Pass Assemblers, Single pass Assembler for Intel x86 ,  
Algorithm of Single Pass Assembler, Multi-Pass Assemblers, Advanced  
Assembly Process, Variants of Assemblers Design of two pass  
assembler, | 06 | 15% |
| 4       | Macro and Macro Processors  
Introduction, Macro Definition and Call, Macro Expansion, Nested  
Macro Calls, Advanced Macro Facilities, Design Of a Macro Pre-  
processor, Design of a Macro Assembler, Functions of a Macro  
Processor, Basic Tasks of a Macro Processor, Design Issues of Macro  
Processors, Features, Macro Processor Design Options, Two-Pass Macro  
Processors, One-Pass Macro Processors | 08 | 20% |
| 5       | Linkers and Loaders | 06 | 20% |
### Suggested Specification table with Marks (Theory):

<table>
<thead>
<tr>
<th>Course Outcome</th>
<th>Distribution of Theory Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Scanning and Parsing</td>
</tr>
<tr>
<td>7</td>
<td>Compilers</td>
</tr>
<tr>
<td>8</td>
<td>Interpreters &amp; Debuggers</td>
</tr>
</tbody>
</table>

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom’s Taxonomy)

Note: This specification table shall be treated as a general guideline for students and teachers. The actual distribution of marks in the question paper may vary slightly from above table.

### Reference Books:

2. System Programming by Srimanta Pal OXFORD Publication

### Course Outcome:

After learning the course the students should be able to:

1. To understand the execution process of HLL programs.
2. To understand the working of scanners and parsers.
3. To understand the basic design of various system software.
4. To implement various system software.

List of Experiments and Design based Problems (DP)/Open Ended Problem:

(Pl. Note: List of Experiments should be as per theory covered in the class, below mentioned practical are just for the reference purpose)

1. Write a program to implement the lexical analyzer.
2. Write a Lexical Analyzer (using lex utility for UNIX).
3. Write a program to left factor the given grammar.
4. Write a program to remove the Left Recursion from a given grammar.
5. Aim: Implement Recursive Descendent Parsing for the given Grammar.
   \[ E : T + E / T \]
   \[ T : F * T / F \]
   \[ F : ( E ) / i \]
6. Implement Predictive Parser for the given grammar.
   \[ E : T + E / T \]
   \[ T : F * T / F \]
   \[ F : ( E ) / i \]
7. Write a SAL program in text file and generate SYMTAB and LITTAB
8. Use macro features of C language
9. Write a program which generates Quadruple Table for the given postfix String
10. Write a C program to parse a given string using Predictive parsing for given grammar.
    \[ \text{type} \rightarrow \text{simple} | \uparrow \text{id} | \text{array} \ [ \text{simple} ] \ \text{of type} \]
    \[ \text{simple} \rightarrow \text{integer} | \text{char} | \text{num dotdot num} \]

List of Open Source Software/learning website:

- www.cs.jhu.edu/~scott/pl/lectures/parsing.html
- www.en.wikipedia.org/wiki/System_programming

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work – The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.