SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

19ECT312 – EMBEDDED SYSTEM DESIGN

III YEAR/ VI SEMESTER

UNIT 3 – PROGRAMMING CONCEPTS AND EMBEDDED PROGRAMMING IN C++

TOPIC 8 – Optimization of Memory Needs

Optimization of Memory Needs

Introduction

- Embedded systems are computing devices designed to perform specific functions, often with constraints on resources like memory, processing power, and energy

- Memory optimization is crucial in embedded system design due to limitations in hardware resources and cost considerations.

Memory Types in Embedded Systems

Different types of memory: RAM, ROM, Flash, EEPROM, etc.

Characteristics and limitations of each type:

RAM: Volatile, fast access, limited size
ROM: Non-volatile, read-only, limited size
Flash: Non-volatile, limited write cycles, slower access than RAM
EEPROM: Non-volatile, electrically erasable, limited write cycles

Memory hierarchy and its implications on system performance and cost

Memory Usage Analysis

- Profiling memory usage: Tools and techniques to analyze memory consumption
- Identifying memory bottlenecks: Areas consuming excessive memory
- Understanding memory fragmentation and its impact

Techniques for Memory Optimization

Data compression techniques:

Lossless compression algorithms like Run-Length Encoding (RLE), Huffman Coding.
Lossy compression techniques for non-critical data.

Code optimization:

Use of efficient algorithms and data structures to minimize code size.
Compiler optimizations: -O flags, dead code elimination, loop unrolling.

Memory pooling and dynamic memory allocation:

Implementing custom memory pools for frequently allocated objects.
Using fixed-size allocation schemes to avoid fragmentation.

Memory-mapped I/O:

Leveraging memory-mapped I/O to reduce RAM usage.
Efficiently managing memory-mapped registers and buffers.

Offloading to external memory:

Utilizing external storage devices like SD cards, EEPROMs for less frequently accessed data.
Strategies for efficient data transfer between internal and external memory.

Virtual memory techniques:

Implementing virtual memory systems for embedded systems with memory constraints.
Page swapping strategies to manage memory overflow.