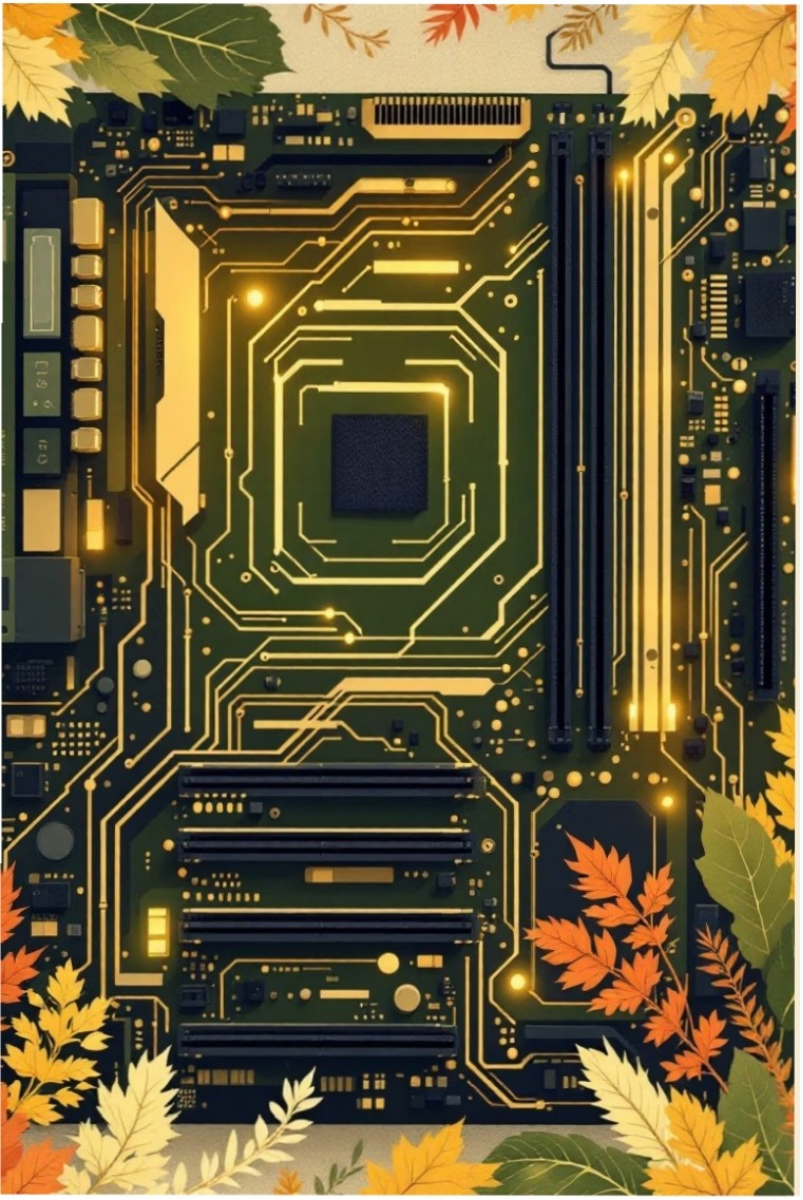

System Resource Info Tool

NitinKumar I B.Tech CSE(AIML)-Semester1
RungtaInternational SkillsUniversity | AcademicYear2025

INTRODUCTION

Understandi ng System Resources in C

What Are System Resources?

Systemresourcesreferto memory allocation and storage capacity available for different data types. In C programming, understanding how much memory each data type consumes is fundamental to writing efficient code.

Why Memory Limits Matter

Everydatatype has specific sizeconstrai nts and value ranges. Knowing these limits helps prevent overflow errors, optimize memory usage,and write portable code across different systems.

Project Overview

C Program Tool

Apractical utility that displays comprehensive information about basic C data types

Memory Analysis

Reveals theexact size in bytes for each data type on your system

Value Limits

Showsminimumand maximum values that each data type can store

This project leverages standard header files (limits.h and float.h) to provide system-dependent information, demonstrating how C programming adapts to different hardware architectures.

Problem Statement

Abstract Concepts

Students often struggle to visualize how data types actually consume memory in the computer system

Hidden Limitations

Without proper tools, understanding the boundaries of integer overflow or floating-point precision remains challenging

System Variability

Datatype sizescan differ across platforms, making it difficult to predict program behavior without concrete information

D This project bridges the gap between theoretical knowledge and practical implementat ion, making abstract memory concepts tangible and measurable.

OBJECTIVE
S

Project Goals & Learning Outcomes

01

Master Data Types

Gain comprehensive understanding of all basic C data types including char, int,float, double, and their variations

02

Analyze Memory Usage

Learn how to measure and interpret memory allocation using the sizeof() operator

03

Explore System Limits

Discover platform-specific constraints and understand portability implications

04

Apply Standard Libraries

Practice using stdio.h, limits.h, and float.h for system-level programm ing

Tools & Technologies

C Language

Coreprogramming language providing low-level memory access and system interaction

VS Code

Modern.lightweight IDE with excellent *CIC++* extension support

Standard Header Files

•stdio.h - Input/output operations

•limits.h - Integer type limits and constants

•float.h - Floating-point type characteristics

How the Program Works

sizeof() Operator

Calculatesmemorys ize in bytes for each data type at compile time

limits.h Constants

Retrieves MINandMAXvalues for integer types (INT_MIN, INT_MAX, etc.)

float.h Macros

10

Accessesfloati ng-point characteristics (FLT_MIN, DBL_MAX, precision)

Console Output

Displaysformattedresu lts using printf() statements

Code Implementation

```
#include <stdio.h>
#include <limits.h>
#include <float.h>

int main() {
printf("=== C Data Type Information ===\n\n");


    // Character types
    printf("char: %zu bytes [%d to %d]\n",
    sizeof(char), CHAR_MIN, CHAR_MAX);

    // Integer types
    printf("int: %zu bytes [%d to %d]\n",
    sizeof(int), INT_ MIN, INT_ MAX);
    printf("long:%zu bytes [%ld to %1d]\n",
    sizeof(long),  LONG_ MIN,LONG_MAX);


    // Floating-point types
    printf("float: %zu bytes [%e to %e]\n",
    sizeof(float),  FLT_ MIN, FLT_MAX);
    printf("double: %zu bytes [%e to %e]\n",
    sizeof(double), DBL_MIN, DBL_MAX);



    return O;
}
```

Key Points: The program includes three essential headers, then systematically displays size and range information for each fundamental data type. The %zu format specifier handles sizeof() results, while %d, %ld, and %e format integers and scientific notation appropriately .

Applications & Adva ntages

Practical Applications

- **Educational Foundation**

  Perfectlearningtoo lforCSfundamenta ls and memory management concepts

- **System Programming**

  Essentialknowledgeforlow-level programming and embedded systems

- **Cross-Platform Development**

  Helpsunderstandpo rtabilitychallenges across different architectures

Key Advantages

Simplicity

Clean,straightforward implementation requiring minimal code

Beginner-Friendly

UsesonlybasicCconcepts, ideal for first-semester students

(Portable

Runsonany system with a standard C compiler

Future Scope & Conclusion

Interactive Input

Add user menuto select specific data types for detailed analysis

File Export

Generatereports saved to text files for documentation purposes

Multi-Language Support

Extendconcept toJava,Python,C++ for comparative analysis

Key Takeaway

This proj ect successfully demonstrates fundamental C programming concepts while providing practical insight into system level memory architecture. It serves as an essential steppingstone for understanding how computers store and manipulate data, preparing students for advanced topics in systems programming and software optimization.