Program Layout & Execution Project

FA25: CMPE-220 Sec 01 - System Software

Instructor: Prof. Ishie Eswar

Semester: Fall 2025

Team Members:
- Priyanshu Patel
- Meet Hitesh Thakkar
- Yash Bharatbhai Savani
- Puneeth Regonda

Date : 3 Dec 2025

GitHub Repository

Project Repository:
https://github.com/priyanshu1044/part2_program_layout

This repository contains the full C code, memory diagrams, recursion demonstration, and project video.

How to Download, Compile, and Run the Program

1. Download the Code

Option A - Using Git:

```
git clone https://github.com/priyanshu1044/part2_program_layout.git
cd part2_program_layout
```

2. Compile the Program

If using the Makefile:

```
make clean && make
```

If compiling manually:

```
gcc main.c recursion.c -o program
```

3. Run the Program

```
./assembler programs/factorial.asm factorial.bin
```

```
./cpu factorial.bin
```

4. Expected Output

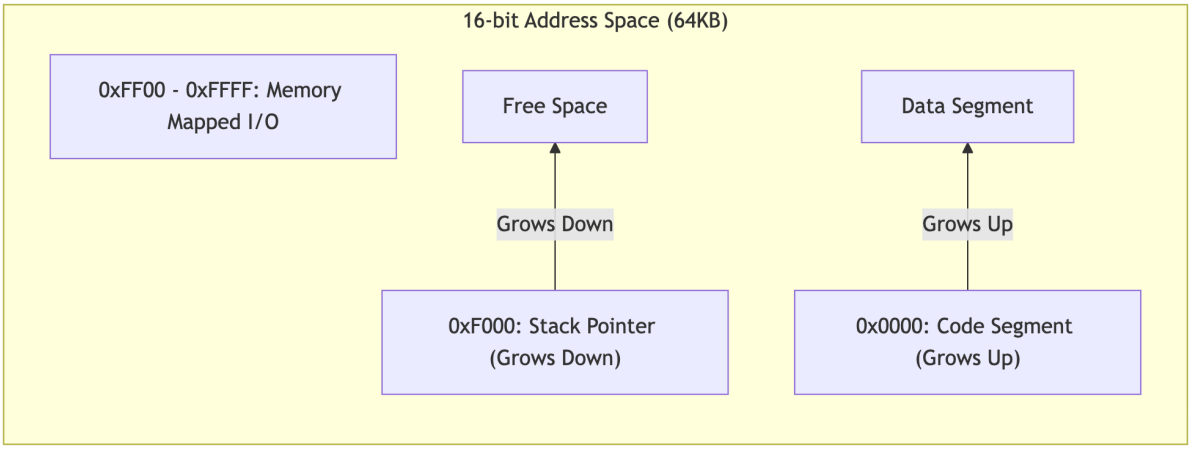
Factorial of 5 = 120

Explanation of Program Layout, Function Calls, Recursion

Program Memory Layout

The program uses a simplified 16-bit CPU model with the following segments:

- Code Segment – contains instructions (grows upward)
- Data Segment – global/static variables
- Free Space – unused memory
- Stack Segment – stores function call frames (grows downward)
- I/O Space – memory-mapped I/O region



Function Call Execution

When main() calls factorial(n):

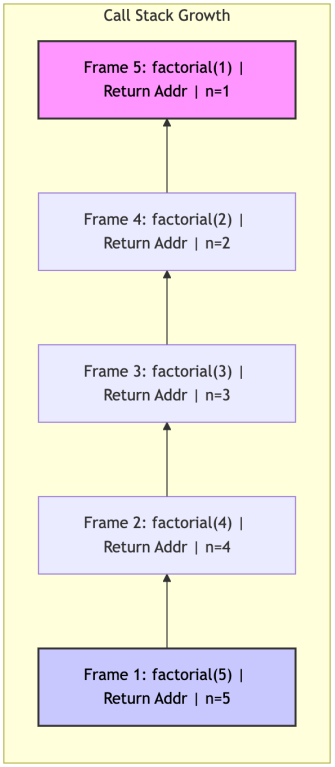
- A stack frame is pushed
- Parameters are stored
- Return address is saved
- CPU jumps to factorial()

How Recursion Works

factorial(5)
→ factorial(4)
→ factorial(3)
→ factorial(2)
→ factorial(1)
→ return 1

Each call pushes a stack frame.
Each return pops a frame and multiplies the result.

Recursion Stack Diagram



Team Member Contributions

Priyanshu Patel:
- Implemented C recursion code
- Wrote factorial() function
- Created memory map diagram

Meet Hitesh Thakkar:
- Created video demonstration
- Explained stack frames and recursion
- Recorded narration

Yash Bharatbhai Savani:
- Organized GitHub repository
- Created Makefile and folder structure
- Completed the final write-up

Puneeth Regonda:
- Developed reference C version
- Verified assembly → C correctness
- Assisted with documentation

Demo Link:

https://github.com/priyanshu1044/part2_program_layout/tree/main/demo-video