

Design and Implementation of a Bootable x86-16 Assembly Operating System

—
A Comprehensive Analysis Report

Course: Operating Systems (CSYE 6230)

Instructor: Prof. Ahmed Banafa

Author: Dennis Sharon Cheruvathoor Shaj

Institution: Northeastern University

Date: December 11, 2025

Purpose of Operating System:

- This OS is a bootable x86-16 real-mode operating system demonstrating fundamental OS development through three NASM assembly modules totaling 325+ lines of MBR-compliant code. The **dennis_os.asm** module implements file system operations (LIST, MOVE, RENAME, DELETE), **dennis_os_memory.asm** provides memory management (ALLOCATE, READ, COPY, CLEAR), and **dennis_os_compute.asm** executes arithmetic operations (ADD, SUB, CMP, MUL). Development required NASM assembler, QEMU x86_64 emulator, MSYS2 environment, with strict 512-byte boot sector constraints and boot signature (0x55AA) compliance for proper bootloader functionality.
- Automated verification via verify_all.sh script validated binary size, boot signatures, and function counts across all modules. Major challenges included boot signature misplacement at offset 512, binary size overflow (514 bytes), stack corruption from improper BP/SP initialization, hexadecimal display logic errors, BIOS INT 0x10 interrupt failures, and position-independent code addressing issues—all successfully resolved through systematic debugging.

Module 1: Dennis OS Main (File System Operations, 122 lines of code)

1. **LIST** - Displays file directory with three files (file1.txt, file2.dat, file3.bin) using print routine via SI register and LODSB instruction
2. **MOVE** - Transfers 16-bit data from memory address 0x9500 to 0x9600 using MOV AX, [0x9500]; MOV [0x9600], AX
3. **RENAME** - Assigns identifier 0xDEAD to memory address 0x9700, simulating file renaming operation
4. **DELETE** - Clears memory at 0x9500 by writing zero value (MOV word [0x9500], 0x0000)

Module 2: Dennis OS Memory Manager (Memory Operations, 108 lines of code)

1. **ALLOCATE (f1)** - Allocates 16-bit value 0xABCD at address 0x9500 using MOV word [0x9500], 0xABCD
2. **READ (f2)** - Reads allocated value from 0x9500 into DX register, displays via hex output routine
3. **COPY (f3)** - Copies data from 0x9500 to backup location 0x9600 using AX as intermediate register
4. **CLEAR (f4)** - Deallocates memory by zeroing address 0x9500 (MOV word [0x9500], 0)

Module 3: Dennis OS Compute Engine (Arithmetic Operations, 95 lines of code)

1. **ADD (f1)** - Performs $0x1234 + 0x5678 = 0x68AC$ using ADD AX, BX instruction, displays all three values in hexadecimal
2. **SUB (f2)** - Performs $0xFFFF - 0x1111 = 0xEEEE$ using SUB AX, BX instruction with immediate display
3. **CMP (f3)** - Compares two equal values ($0xABCD == 0xABCD$) using CMP instruction with conditional jump to display "EQUAL" or "NOT EQ"
4. **MUL (f4)** - Multiplies by shifting ($0x0010 \ll 1 = 0x0020$) using SHL AX, 1 instruction, demonstrates bitwise operations

- **Total: 325+ lines of assembly code**