Intel 3003 emulator

Wednesday, 2 April 2025

10:20 AM

**Intel 3003: History, Function, and Instruction Set Overview**

The **Intel 3003** is one of Intel's earliest microprocessors, introduced in 1972. It was designed and developed as a part of Intel's initial foray into the microprocessor market, which would later revolutionize the computing industry. While the Intel 3003 might not have the widespread recognition of later Intel processors, it played a crucial role in the development of microprocessor technology. In this essay, we will explore the history, function, and the instruction set of the Intel 3003, along with a brief look at its legacy in the broader context of computing history.

**1. Historical Background**

The Intel 3003 was a product of a time when the semiconductor industry was still in its infancy. It came out during the early 1970s, a period when computing was transitioning from large, room-sized mainframes to more compact systems. The Intel 3003 was an early 16-bit processor, and it was primarily targeted at the embedded systems market. While Intel had already released the famous **Intel 4004**, the 3003 had a smaller, more specialized role, focusing on specific tasks rather than general-purpose computing.

The 3003 processor was built on the foundation of Intel's previous successes but was designed for more limited applications. Its development marked Intel's growing presence in the microprocessor industry, which would later dominate computing with subsequent releases such as the Intel 8080 and 8086.

Intel's move into the microprocessor market was in direct response to the increasing demand for smaller, more efficient computing devices. The Intel 3003, while not widely used, laid the groundwork for Intel’s future breakthroughs in computing.

**2. Function and Architecture**

The Intel 3003 microprocessor was an **8-bit CPU**, meaning it could process 8 bits of data in a single cycle. The chip had a relatively simple architecture compared to modern processors, featuring basic arithmetic and logical operations. It was designed to perform specific control tasks in embedded systems, particularly in applications that required relatively low computational power but high reliability.

Key features of the Intel 3003's function include:

* **16-bit Data Bus**: The Intel 3003 had an 16-bit data bus, which allowed it to handle 16-bit wide instructions and data. This was a step up from earlier designs, allowing for more complex operations than earlier processors like the Intel 4004.
* **Direct Addressing**: The processor supported direct addressing of memory, making it easier to access specific data in memory locations without complex intermediary steps.
* **Control and Timing**: The Intel 3003 was optimized for control-oriented tasks. It was capable of executing sequences of instructions for controlling embedded systems, such as handling input/output tasks in electronics.

Despite its capabilities, the Intel 3003 did not have the raw computational power of later processors. Its primary function was embedded control, such as managing the flow of data in specific applications rather than general-purpose computing.

**3. Instruction Set and Programming Model**

The instruction set of the Intel 3003 was relatively simple compared to later microprocessors. Being an early 8-bit processor, it did not have the advanced instruction sets of more complex CPUs. Its instruction set was specifically tailored to the needs of embedded applications, allowing it to execute basic operations in a straightforward manner.

**Instruction Set Overview**

The Intel 3003 featured a limited but functional instruction set, typical for processors of its era. The key operations it could perform included:

1. **Arithmetic Instructions**:
   * **ADD: Add two operands.**
   * **SUB: Subtract two operands.**
   * **INC: Increment an operand by one.**
   * **DEC: Decrement an operand by one.**
2. **Logical Instructions**:
   * **AND: Perform a logical AND on two operands.**
   * **OR: Perform a logical OR on two operands.**
   * **XOR: Perform a logical XOR on two operands.**
3. **Control Instructions**:
   * **JMP: Jump to a specific address in memory (unconditional branch).**
   * **JC: Jump if the carry flag is set.**
   * **JZ: Jump if the zero flag is set.**
4. **Data Movement Instructions**:
   * **MOV: Move data from one register or memory location to another.**
   * **PUSH: Push data onto the stack.**
   * **POP: Pop data from the stack.**
5. **Bitwise Operations**:
   * **SHL: Shift left (multiply by 2).**
   * **SHR: Shift right (divide by 2).**
6. **Flags and Status**:
   * **The processor featured status flags, such as carry, zero, and sign, to track the result of operations. These flags influenced conditional jump instructions and were essential for controlling program flow.**

**Programming Model**

Programming the Intel 3003 required low-level access to hardware, using assembly language or machine code. Each instruction was executed in a specific sequence, with the results of each operation affecting the processor's internal registers and flags. The simplicity of the instruction set made it accessible for embedded systems developers, though it was not designed for general-purpose computing tasks like text processing or complex calculations.

**4. Legacy and Impact**

Though the Intel 3003 was not a mainstream processor in terms of market impact, its existence was important in the early development of microprocessors. As one of Intel's first 16-bit CPUs, it helped pave the way for more advanced chips. While it was not used for general-purpose computing, its role in embedded systems was significant, especially for applications in industrial control and basic automation.

The Intel 3003's legacy can be seen in the subsequent Intel microprocessors that dominated the 1970s and 1980s, such as the **Intel 8080** and **Intel 8086**, which formed the foundation for the x86 architecture still in use today. The simplicity and efficiency of the 3003’s instruction set influenced future designs that prioritized ease of use in embedded environments.

In conclusion, the Intel 3003 may not be as well-known as later processors in Intel’s lineup, but it represents a significant stepping stone in the early days of microprocessor history. Its role in embedded systems and the fundamental instruction set it utilized helped shape the future of microprocessor design, with lasting impacts on the development of computing technologies.

V1.1 C++ (x86-16bit 2 core 3003 emulator)

Instruction set and accepted commands:

AND XAND NAND XNAND

OR XOR NOR XNOR

NOT XNOT

POP POPF

JMP JC JZ

ADD SUB  
MUL DIV

DEC INC

MOV

SHL SHR  
LOCK

XCHG

CMPXCHG

PUSH

CLI STI

FWAIT

SFENCE LFENCE MFENCE

Emulator C++

#include <iostream>

#include <bitset>

#include <vector>

#include <thread>

#include <mutex>

std::mutex core\_mutex;

// Basic logic gates

bool AND(bool a, bool b) { return a & b; }

bool OR(bool a, bool b) { return a | b; }

bool XOR(bool a, bool b) { return a ^ b; }

bool NOT(bool a) { return !a; }

bool NAND(bool a, bool b) { return !(a & b); }

bool NOR(bool a, bool b) { return !(a | b); }

bool XNOR(bool a, bool b) { return !(a ^ b); }

bool NXOR(bool a, bool b) { return !(a ^ b); }

bool XAND(bool a, bool b) { return a & !b; }

bool NXAND(bool a, bool b) { return !(a & !b); }

bool XNOT(bool a) { return !a; }

// Half Adder

void halfAdder(bool a, bool b, bool &sum, bool &carry) {

sum = XOR(a, b);

carry = AND(a, b);

}

// Full Adder

void fullAdder(bool a, bool b, bool cin, bool &sum, bool &carry) {

bool s1, c1, c2;

halfAdder(a, b, s1, c1);

halfAdder(s1, cin, sum, c2);

carry = OR(c1, c2);

}

// 16-bit ALU (Add/Subtract/Multiply/Divide)

std::bitset<16> ALU(std::bitset<16> A, std::bitset<16> B, bool subtract, bool multiply, bool divide) {

std::bitset<16> result;

bool carry = subtract;

if (multiply) {

result = std::bitset<16>(A.to\_ulong() \* B.to\_ulong());

} else if (divide) {

if (B.to\_ulong() != 0)

result = std::bitset<16>(A.to\_ulong() / B.to\_ulong());

else

result = 0; // Prevent division by zero

} else {

for (int i = 0; i < 16; i++) {

bool sum, c;

fullAdder(A[i], XOR(B[i], subtract), carry, sum, c);

result[i] = sum;

carry = c;

}

}

return result;

}

// Increment & Decrement

std::bitset<16> INC(std::bitset<16> A) {

return ALU(A, std::bitset<16>(1), false, false, false);

}

std::bitset<16> DEC(std::bitset<16> A) {

return ALU(A, std::bitset<16>(1), true, false, false);

}

// Shift Left and Shift Right

std::bitset<16> SHL(std::bitset<16> A) {

return A << 1;

}

std::bitset<16> SHR(std::bitset<16> A) {

return A >> 1;

}

// Registers and Memory

struct CPU {

std::bitset<16> registers[8]; // 8 general-purpose registers

std::vector<std::bitset<16>> memory;

// Stack pointer and flags

int sp = 0; // Stack pointer for PUSH/POP operations

bool zeroFlag = false;

bool carryFlag = false;

CPU(size\_t memory\_size) : memory(memory\_size) {}

// MOV instruction: Move value into register

void MOV(int reg, std::bitset<16> value) {

registers[reg] = value;

}

// JMP instruction: Jump to specific memory address (unconditional)

void JMP(int address) {

sp = address; // Set stack pointer to the jump address (simulating jump)

}

// JC instruction: Jump if carry flag is set

bool JC() {

return carryFlag; // Jump if carry flag is set

}

// JZ instruction: Jump if zero flag is set

bool JZ() {

return zeroFlag; // Jump if zero flag is set

}

// PUSH instruction: Push register value to stack

void PUSH(int reg) {

if (sp < memory.size()) {

memory[sp] = registers[reg]; // Store value in memory

sp++;

}

}

// POP instruction: Pop value from stack into register

void POP(int reg) {

if (sp > 0) {

sp--;

registers[reg] = memory[sp]; // Retrieve value from memory

}

}

// SUB instruction: Subtract register values

void SUB(int reg1, int reg2) {

registers[reg1] = ALU(registers[reg1], registers[reg2], true, false, false);

}

// ADD instruction: Add register values

void ADD(int reg1, int reg2) {

registers[reg1] = ALU(registers[reg1], registers[reg2], false, false, false);

}

// Lock Instruction: For multi-core synchronization

void LOCK() {

std::lock\_guard<std::mutex> lock(core\_mutex);

}

// XCHG Instruction: Exchange register values (for atomic operations)

void XCHG(int reg1, int reg2) {

std::bitset<16> temp = registers[reg1];

registers[reg1] = registers[reg2];

registers[reg2] = temp;

}

// CMPXCHG Instruction: Compare and exchange register values

void CMPXCHG(int reg1, int reg2) {

if (registers[reg1] == registers[reg2]) {

registers[reg1] = registers[reg2]; // Exchange if equal

}

}

// CLI Instruction: Clear interrupt flag

void CLI() {

// In a real CPU, this would disable interrupts.

std::cout << "Interrupt flag cleared.\n";

}

// STI Instruction: Set interrupt flag

void STI() {

// In a real CPU, this would enable interrupts.

std::cout << "Interrupt flag set.\n";

}

// SFENCE Instruction: Memory fence to ensure stores are ordered

void SFENCE() {

// In a real CPU, this would enforce memory ordering between cores.

std::cout << "Store fence applied.\n";

}

// LFENCE Instruction: Memory fence to ensure loads are ordered

void LFENCE() {

// In a real CPU, this would enforce load ordering between cores.

std::cout << "Load fence applied.\n";

}

// MFENCE Instruction: Memory fence to ensure all memory operations are ordered

void MFENCE() {

// In a real CPU, this would ensure that all memory operations are complete before proceeding.

std::cout << "Memory fence applied.\n";

}

};

// Simulate Execution for a Core

void execute\_core(CPU &cpu, int core\_id) {

std::lock\_guard<std::mutex> lock(core\_mutex);

std::cout << "Executing on core " << core\_id << "\n";

// MOV instruction example

cpu.MOV(0, std::bitset<16>("0000000000001010")); // MOV 10 to register 0

cpu.MOV(1, std::bitset<16>("0000000000000101")); // MOV 5 to register 1

cpu.ADD(0, 1); // ADD register 1 to register 0 (10 + 5)

cpu.PUSH(0); // PUSH register 0 onto the stack

cpu.POP(2); // POP the value into register 2

cpu.SUB(2, 1); // SUB register 1 from register 2 (result should be 0)

// Test atomic exchange and compare exchange

cpu.XCHG(0, 1);

cpu.CMPXCHG(0, 1);

}

// Main function

int main() {

CPU core1(1024), core2(1024);

std::bitset<16> A("0000000000001010"); // 10

std::bitset<16> B("0000000000000101"); // 5

// ALU operations

std::cout << "A + B = " << ALU(A, B, false, false, false) << "\n";

std::cout << "A - B = " << ALU(A, B, true, false, false) << "\n";

std::cout << "A \* B = " << ALU(A, B, false, true, false) << "\n";

std::cout << "A / B = " << ALU(A, B, false, false, true) << "\n";

std::cout << "INC(A) = " << INC(A) << "\n";

std::cout << "DEC(A) = " << DEC(A) << "\n";

// Shift operations

std::cout << "SHL(A) = " << SHL(A) << "\n";

std::cout << "SHR(A) = " << SHR(A) << "\n";

// Multi-core execution

std::thread t1(execute\_core, std::ref(core1), 1);

std::thread t2(execute\_core, std::ref(core2), 2);

t1.join();

t2.join();

// Example of using conditional jump (JZ and JC)

if (core1.JZ()) {

std::cout << "Jump to address (Zero flag is set)\n";

}

if (core1.JC()) {

std::cout << "Jump to address (Carry flag is set)\n";

}

return 0;

}

**3003 x86-16 multicore instruction set reference table:**

|  |  |
| --- | --- |
| **Instruction** | **Description** |
| **AND** | Performs a bitwise AND operation between two operands. |
| **XAND** | Exclusive AND, performs a bitwise AND operation only where both operands are 1. |
| **NAND** | Performs a bitwise NAND operation (NOT AND). |
| **XNAND** | Exclusive NAND, the inverse of Exclusive AND. |
| **OR** | Performs a bitwise OR operation between two operands. |
| **XOR** | Performs a bitwise XOR (exclusive OR) operation between two operands. |
| **NOR** | Performs a bitwise NOR operation (NOT OR). |
| **XNOR** | Exclusive NOR, the inverse of XOR (if operands are the same, result is 1). |
| **NOT** | Performs a bitwise NOT operation (inverts all bits of the operand). |
| **XNOT** | Exclusive NOT, a variation that works with a single operand, but with exclusive logic. |
| **POP** | Pops the top value from the stack into a register or memory. |
| **JMP** | Jump to a specified address or label unconditionally. |
| **JC** | Jump if carry flag is set (used for conditional branching based on the carry flag). |
| **JZ** | Jump if zero flag is set (used for conditional branching based on the zero flag). |
| **ADD** | Adds two operands together and stores the result in the destination. |
| **SUB** | Subtracts the second operand from the first operand and stores the result in the destination. |
| **MUL** | Unsigned multiplication of the accumulator register by a specified operand. Result is stored across two registers (AX, DX). |
| **DIV** | Unsigned division, divides the value in AX by the operand. Quotient is stored in AX, remainder in DX. |
| **DEC** | Decrements the value of a register or memory operand by 1. |
| **INC** | Increments the value of a register or memory operand by 1. |
| **MOV** | Moves (copies) data from one register/memory location to another. |
| **SHL** | Shift left, shifts bits of the operand to the left, inserting zeroes into the rightmost bits. |
| **SHR** | Shift right, shifts bits of the operand to the right, inserting zeroes into the leftmost bits. |
| **LOCK** | Used to acquire a lock on a memory location, ensuring atomicity for multi-threaded operations. |
| **XCHG** | Exchanges the contents of two registers or a register and memory location. Often used in synchronization. |
| **CMPXCHG** | Compares two operands, and if they are equal, replaces the value in memory with the second operand. Often used for atomic operations. |
| **PUSH** | Pushes a value onto the stack, used in multi-threaded environments for saving state. |
| **POPF** | Pops the flags register off the stack, restoring its state. Can be used to save/restore processor state in multi-threading. |
| **CLI** | Clears the interrupt flag (disables interrupts), often used in multi-core systems to prevent interruptions during critical code sections. |
| **STI** | Sets the interrupt flag (enables interrupts), typically used after critical sections of code in multi-core systems to allow interrupt handling again. |
| **FWAIT** | Waits for an FPU (Floating Point Unit) operation to finish, used in synchronization in multi-threaded contexts. |
| **SFENCE** | Ensures that all previous store operations to memory are completed before the fence, useful for memory ordering in multi-core systems. |
| **LFENCE** | Ensures that all previous load operations are completed before the fence, used for synchronization in multi-threaded contexts. |
| **MFENCE** | Ensures that all previous memory operations (loads and stores) are completed before the fence. It’s used to enforce ordering in multi-core systems. |

x86 dump

asm:

0: 23 69 6e and ebp,DWORD PTR [ecx+0x6e]

3: 63 6c 75 64 arpl WORD PTR [ebp+esi\*2+0x64],bp

7: 65 20 3c 69 and BYTE PTR gs:[ecx+ebp\*2],bh

b: 6f outs dx,DWORD PTR ds:[esi]

c: 73 74 jae 0x82

e: 72 65 jb 0x75

10: 61 popa

11: 6d ins DWORD PTR es:[edi],dx

12: 3e 0a 23 or ah,BYTE PTR ds:[ebx]

15: 69 6e 63 6c 75 64 65 imul ebp,DWORD PTR [esi+0x63],0x6564756c

1c: 20 3c 62 and BYTE PTR [edx+eiz\*2]a,bh

1f: 69 74 73 65 74 3e 0a imul esi,DWORD PTR [ebx+esi\*2+0x65],0x230a3e74

26: 23

27: 69 6e 63 6c 75 64 65 imul ebp,DWORD PTR [esi+0x63],0x6564756c

2e: 20 3c 76 and BYTE PTR [esi+esi\*2],bh

31: 65 63 74 6f 72 arpl WORD PTR gs:[edi+ebp\*2+0x72],si

36: 3e 0a 23 or ah,BYTE PTR ds:[ebx]

39: 69 6e 63 6c 75 64 65 imul ebp,DWORD PTR [esi+0x63],0x6564756c

40: 20 3c 74 and BYTE PTR [esp+esi\*2],bh

43: 68 72 65 61 64 push 0x64616572

48: 3e 0a 23 or ah,BYTE PTR ds:[ebx]

4b: 69 6e 63 6c 75 64 65 imul ebp,DWORD PTR [esi+0x63],0x6564756c

52: 20 3c 6d 75 74 65 78 and BYTE PTR [ebp\*2+0x78657475],bh

59: 3e 0a 0a or cl,BYTE PTR ds:[edx]

5c: 73 74 jae 0xd2

5e: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

61: 6d ins DWORD PTR es:[edi],dx

62: 75 74 jne 0xd8

64: 65 78 20 gs js 0x87

67: 63 6f 72 arpl WORD PTR [edi+0x72],bp

6a: 65 5f gs pop edi

6c: 6d ins DWORD PTR es:[edi],dx

6d: 75 74 jne 0xe3

6f: 65 78 3b gs js 0xad

72: 0a 0a or cl,BYTE PTR [edx]

74: 2f das

75: 2f das

76: 20 42 61 and BYTE PTR [edx+0x61],al

79: 73 69 jae 0xe4

7b: 63 20 arpl WORD PTR [eax],sp

7d: 6c ins BYTE PTR es:[edi],dx

7e: 6f outs dx,DWORD PTR ds:[esi]

7f: 67 69 63 20 67 61 74 imul esp,DWORD PTR [bp+di+0x20],0x65746167

86: 65

87: 73 0a jae 0x93

89: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

8c: 6c ins BYTE PTR es:[edi],dx

8d: 20 41 4e and BYTE PTR [ecx+0x4e],al

90: 44 inc esp

91: 28 62 6f sub BYTE PTR [edx+0x6f],ah

94: 6f outs dx,DWORD PTR ds:[esi]

95: 6c ins BYTE PTR es:[edi],dx

96: 20 61 2c and BYTE PTR [ecx+0x2c],ah

99: 20 62 6f and BYTE PTR [edx+0x6f],ah

9c: 6f outs dx,DWORD PTR ds:[esi]

9d: 6c ins BYTE PTR es:[edi],dx

9e: 20 62 29 and BYTE PTR [edx+0x29],ah

a1: 20 7b 20 and BYTE PTR [ebx+0x20],bh

a4: 72 65 jb 0x10b

a6: 74 75 je 0x11d

a8: 72 6e jb 0x118

aa: 20 61 20 and BYTE PTR [ecx+0x20],ah

ad: 26 20 62 3b and BYTE PTR es:[edx+0x3b],ah

b1: 20 7d 0a and BYTE PTR [ebp+0xa],bh

b4: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

b7: 6c ins BYTE PTR es:[edi],dx

b8: 20 4f 52 and BYTE PTR [edi+0x52],cl

bb: 28 62 6f sub BYTE PTR [edx+0x6f],ah

be: 6f outs dx,DWORD PTR ds:[esi]

bf: 6c ins BYTE PTR es:[edi],dx

c0: 20 61 2c and BYTE PTR [ecx+0x2c],ah

c3: 20 62 6f and BYTE PTR [edx+0x6f],ah

c6: 6f outs dx,DWORD PTR ds:[esi]

c7: 6c ins BYTE PTR es:[edi],dx

c8: 20 62 29 and BYTE PTR [edx+0x29],ah

cb: 20 7b 20 and BYTE PTR [ebx+0x20],bh

ce: 72 65 jb 0x135

d0: 74 75 je 0x147

d2: 72 6e jb 0x142

d4: 20 61 20 and BYTE PTR [ecx+0x20],ah

d7: 7c 20 jl 0xf9

d9: 62 3b bound edi,QWORD PTR [ebx]

db: 20 7d 0a and BYTE PTR [ebp+0xa],bh

de: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

e1: 6c ins BYTE PTR es:[edi],dx

e2: 20 58 4f and BYTE PTR [eax+0x4f],bl

e5: 52 push edx

e6: 28 62 6f sub BYTE PTR [edx+0x6f],ah

e9: 6f outs dx,DWORD PTR ds:[esi]

ea: 6c ins BYTE PTR es:[edi],dx

eb: 20 61 2c and BYTE PTR [ecx+0x2c],ah

ee: 20 62 6f and BYTE PTR [edx+0x6f],ah

f1: 6f outs dx,DWORD PTR ds:[esi]

f2: 6c ins BYTE PTR es:[edi],dx

f3: 20 62 29 and BYTE PTR [edx+0x29],ah

f6: 20 7b 20 and BYTE PTR [ebx+0x20],bh

f9: 72 65 jb 0x160

fb: 74 75 je 0x172

fd: 72 6e jb 0x16d

ff: 20 61 20 and BYTE PTR [ecx+0x20],ah

102: 5e pop esi

103: 20 62 3b and BYTE PTR [edx+0x3b],ah

106: 20 7d 0a and BYTE PTR [ebp+0xa],bh

109: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

10c: 6c ins BYTE PTR es:[edi],dx

10d: 20 4e 4f and BYTE PTR [esi+0x4f],cl

110: 54 push esp

111: 28 62 6f sub BYTE PTR [edx+0x6f],ah

114: 6f outs dx,DWORD PTR ds:[esi]

115: 6c ins BYTE PTR es:[edi],dx

116: 20 61 29 and BYTE PTR [ecx+0x29],ah

119: 20 7b 20 and BYTE PTR [ebx+0x20],bh

11c: 72 65 jb 0x183

11e: 74 75 je 0x195

120: 72 6e jb 0x190

122: 20 21 and BYTE PTR [ecx],ah

124: 61 popa

125: 3b 20 cmp esp,DWORD PTR [eax]

127: 7d 0a jge 0x133

129: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

12c: 6c ins BYTE PTR es:[edi],dx

12d: 20 4e 41 and BYTE PTR [esi+0x41],cl

130: 4e dec esi

131: 44 inc esp

132: 28 62 6f sub BYTE PTR [edx+0x6f],ah

135: 6f outs dx,DWORD PTR ds:[esi]

136: 6c ins BYTE PTR es:[edi],dx

137: 20 61 2c and BYTE PTR [ecx+0x2c],ah

13a: 20 62 6f and BYTE PTR [edx+0x6f],ah

13d: 6f outs dx,DWORD PTR ds:[esi]

13e: 6c ins BYTE PTR es:[edi],dx

13f: 20 62 29 and BYTE PTR [edx+0x29],ah

142: 20 7b 20 and BYTE PTR [ebx+0x20],bh

145: 72 65 jb 0x1ac

147: 74 75 je 0x1be

149: 72 6e jb 0x1b9

14b: 20 21 and BYTE PTR [ecx],ah

14d: 28 61 20 sub BYTE PTR [ecx+0x20],ah

150: 26 20 62 29 and BYTE PTR es:[edx+0x29],ah

154: 3b 20 cmp esp,DWORD PTR [eax]

156: 7d 0a jge 0x162

158: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

15b: 6c ins BYTE PTR es:[edi],dx

15c: 20 4e 4f and BYTE PTR [esi+0x4f],cl

15f: 52 push edx

160: 28 62 6f sub BYTE PTR [edx+0x6f],ah

163: 6f outs dx,DWORD PTR ds:[esi]

164: 6c ins BYTE PTR es:[edi],dx

165: 20 61 2c and BYTE PTR [ecx+0x2c],ah

168: 20 62 6f and BYTE PTR [edx+0x6f],ah

16b: 6f outs dx,DWORD PTR ds:[esi]

16c: 6c ins BYTE PTR es:[edi],dx

16d: 20 62 29 and BYTE PTR [edx+0x29],ah

170: 20 7b 20 and BYTE PTR [ebx+0x20],bh

173: 72 65 jb 0x1da

175: 74 75 je 0x1ec

177: 72 6e jb 0x1e7

179: 20 21 and BYTE PTR [ecx],ah

17b: 28 61 20 sub BYTE PTR [ecx+0x20],ah

17e: 7c 20 jl 0x1a0

180: 62 29 bound ebp,QWORD PTR [ecx]

182: 3b 20 cmp esp,DWORD PTR [eax]

184: 7d 0a jge 0x190

186: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

189: 6c ins BYTE PTR es:[edi],dx

18a: 20 58 4e and BYTE PTR [eax+0x4e],bl

18d: 4f dec edi

18e: 52 push edx

18f: 28 62 6f sub BYTE PTR [edx+0x6f],ah

192: 6f outs dx,DWORD PTR ds:[esi]

193: 6c ins BYTE PTR es:[edi],dx

194: 20 61 2c and BYTE PTR [ecx+0x2c],ah

197: 20 62 6f and BYTE PTR [edx+0x6f],ah

19a: 6f outs dx,DWORD PTR ds:[esi]

19b: 6c ins BYTE PTR es:[edi],dx

19c: 20 62 29 and BYTE PTR [edx+0x29],ah

19f: 20 7b 20 and BYTE PTR [ebx+0x20],bh

1a2: 72 65 jb 0x209

1a4: 74 75 je 0x21b

1a6: 72 6e jb 0x216

1a8: 20 21 and BYTE PTR [ecx],ah

1aa: 28 61 20 sub BYTE PTR [ecx+0x20],ah

1ad: 5e pop esi

1ae: 20 62 29 and BYTE PTR [edx+0x29],ah

1b1: 3b 20 cmp esp,DWORD PTR [eax]

1b3: 7d 0a jge 0x1bf

1b5: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

1b8: 6c ins BYTE PTR es:[edi],dx

1b9: 20 4e 58 and BYTE PTR [esi+0x58],cl

1bc: 4f dec edi

1bd: 52 push edx

1be: 28 62 6f sub BYTE PTR [edx+0x6f],ah

1c1: 6f outs dx,DWORD PTR ds:[esi]

1c2: 6c ins BYTE PTR es:[edi],dx

1c3: 20 61 2c and BYTE PTR [ecx+0x2c],ah

1c6: 20 62 6f and BYTE PTR [edx+0x6f],ah

1c9: 6f outs dx,DWORD PTR ds:[esi]

1ca: 6c ins BYTE PTR es:[edi],dx

1cb: 20 62 29 and BYTE PTR [edx+0x29],ah

1ce: 20 7b 20 and BYTE PTR [ebx+0x20],bh

1d1: 72 65 jb 0x238

1d3: 74 75 je 0x24a

1d5: 72 6e jb 0x245

1d7: 20 21 and BYTE PTR [ecx],ah

1d9: 28 61 20 sub BYTE PTR [ecx+0x20],ah

1dc: 5e pop esi

1dd: 20 62 29 and BYTE PTR [edx+0x29],ah

1e0: 3b 20 cmp esp,DWORD PTR [eax]

1e2: 7d 0a jge 0x1ee

1e4: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

1e7: 6c ins BYTE PTR es:[edi],dx

1e8: 20 58 41 and BYTE PTR [eax+0x41],bl

1eb: 4e dec esi

1ec: 44 inc esp

1ed: 28 62 6f sub BYTE PTR [edx+0x6f],ah

1f0: 6f outs dx,DWORD PTR ds:[esi]

1f1: 6c ins BYTE PTR es:[edi],dx

1f2: 20 61 2c and BYTE PTR [ecx+0x2c],ah

1f5: 20 62 6f and BYTE PTR [edx+0x6f],ah

1f8: 6f outs dx,DWORD PTR ds:[esi]

1f9: 6c ins BYTE PTR es:[edi],dx

1fa: 20 62 29 and BYTE PTR [edx+0x29],ah

1fd: 20 7b 20 and BYTE PTR [ebx+0x20],bh

200: 72 65 jb 0x267

202: 74 75 je 0x279

204: 72 6e jb 0x274

206: 20 61 20 and BYTE PTR [ecx+0x20],ah

209: 26 20 21 and BYTE PTR es:[ecx],ah

20c: 62 3b bound edi,QWORD PTR [ebx]

20e: 20 7d 0a and BYTE PTR [ebp+0xa],bh

211: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

214: 6c ins BYTE PTR es:[edi],dx

215: 20 4e 58 and BYTE PTR [esi+0x58],cl

218: 41 inc ecx

219: 4e dec esi

21a: 44 inc esp

21b: 28 62 6f sub BYTE PTR [edx+0x6f],ah

21e: 6f outs dx,DWORD PTR ds:[esi]

21f: 6c ins BYTE PTR es:[edi],dx

220: 20 61 2c and BYTE PTR [ecx+0x2c],ah

223: 20 62 6f and BYTE PTR [edx+0x6f],ah

226: 6f outs dx,DWORD PTR ds:[esi]

227: 6c ins BYTE PTR es:[edi],dx

228: 20 62 29 and BYTE PTR [edx+0x29],ah

22b: 20 7b 20 and BYTE PTR [ebx+0x20],bh

22e: 72 65 jb 0x295

230: 74 75 je 0x2a7

232: 72 6e jb 0x2a2

234: 20 21 and BYTE PTR [ecx],ah

236: 28 61 20 sub BYTE PTR [ecx+0x20],ah

239: 26 20 21 and BYTE PTR es:[ecx],ah

23c: 62 29 bound ebp,QWORD PTR [ecx]

23e: 3b 20 cmp esp,DWORD PTR [eax]

240: 7d 0a jge 0x24c

242: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

245: 6c ins BYTE PTR es:[edi],dx

246: 20 58 4e and BYTE PTR [eax+0x4e],bl

249: 4f dec edi

24a: 54 push esp

24b: 28 62 6f sub BYTE PTR [edx+0x6f],ah

24e: 6f outs dx,DWORD PTR ds:[esi]

24f: 6c ins BYTE PTR es:[edi],dx

250: 20 61 29 and BYTE PTR [ecx+0x29],ah

253: 20 7b 20 and BYTE PTR [ebx+0x20],bh

256: 72 65 jb 0x2bd

258: 74 75 je 0x2cf

25a: 72 6e jb 0x2ca

25c: 20 21 and BYTE PTR [ecx],ah

25e: 61 popa

25f: 3b 20 cmp esp,DWORD PTR [eax]

261: 7d 0a jge 0x26d

263: 0a 2f or ch,BYTE PTR [edi]

265: 2f das

266: 20 48 61 and BYTE PTR [eax+0x61],cl

269: 6c ins BYTE PTR es:[edi],dx

26a: 66 20 41 64 data16 and BYTE PTR [ecx+0x64],al

26e: 64 65 72 0a fs gs jb 0x27c

272: 76 6f jbe 0x2e3

274: 69 64 20 68 61 6c 66 imul esp,DWORD PTR [eax+eiz\*1+0x68],0x41666c61

27b: 41

27c: 64 64 65 72 28 fs fs gs jb 0x2a9

281: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

284: 6c ins BYTE PTR es:[edi],dx

285: 20 61 2c and BYTE PTR [ecx+0x2c],ah

288: 20 62 6f and BYTE PTR [edx+0x6f],ah

28b: 6f outs dx,DWORD PTR ds:[esi]

28c: 6c ins BYTE PTR es:[edi],dx

28d: 20 62 2c and BYTE PTR [edx+0x2c],ah

290: 20 62 6f and BYTE PTR [edx+0x6f],ah

293: 6f outs dx,DWORD PTR ds:[esi]

294: 6c ins BYTE PTR es:[edi],dx

295: 20 26 and BYTE PTR [esi],ah

297: 73 75 jae 0x30e

299: 6d ins DWORD PTR es:[edi],dx

29a: 2c 20 sub al,0x20

29c: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

29f: 6c ins BYTE PTR es:[edi],dx

2a0: 20 26 and BYTE PTR [esi],ah

2a2: 63 61 72 arpl WORD PTR [ecx+0x72],sp

2a5: 72 79 jb 0x320

2a7: 29 20 sub DWORD PTR [eax],esp

2a9: 7b 0a jnp 0x2b5

2ab: 20 20 and BYTE PTR [eax],ah

2ad: 20 20 and BYTE PTR [eax],ah

2af: 73 75 jae 0x326

2b1: 6d ins DWORD PTR es:[edi],dx

2b2: 20 3d 20 58 4f 52 and BYTE PTR ds:0x524f5820,bh

2b8: 28 61 2c sub BYTE PTR [ecx+0x2c],ah

2bb: 20 62 29 and BYTE PTR [edx+0x29],ah

2be: 3b 0a cmp ecx,DWORD PTR [edx]

2c0: 20 20 and BYTE PTR [eax],ah

2c2: 20 20 and BYTE PTR [eax],ah

2c4: 63 61 72 arpl WORD PTR [ecx+0x72],sp

2c7: 72 79 jb 0x342

2c9: 20 3d 20 41 4e 44 and BYTE PTR ds:0x444e4120,bh

2cf: 28 61 2c sub BYTE PTR [ecx+0x2c],ah

2d2: 20 62 29 and BYTE PTR [edx+0x29],ah

2d5: 3b 0a cmp ecx,DWORD PTR [edx]

2d7: 7d 0a jge 0x2e3

2d9: 0a 2f or ch,BYTE PTR [edi]

2db: 2f das

2dc: 20 46 75 and BYTE PTR [esi+0x75],al

2df: 6c ins BYTE PTR es:[edi],dx

2e0: 6c ins BYTE PTR es:[edi],dx

2e1: 20 41 64 and BYTE PTR [ecx+0x64],al

2e4: 64 65 72 0a fs gs jb 0x2f2

2e8: 76 6f jbe 0x359

2ea: 69 64 20 66 75 6c 6c imul esp,DWORD PTR [eax+eiz\*1+0x66],0x416c6c75

2f1: 41

2f2: 64 64 65 72 28 fs fs gs jb 0x31f

2f7: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

2fa: 6c ins BYTE PTR es:[edi],dx

2fb: 20 61 2c and BYTE PTR [ecx+0x2c],ah

2fe: 20 62 6f and BYTE PTR [edx+0x6f],ah

301: 6f outs dx,DWORD PTR ds:[esi]

302: 6c ins BYTE PTR es:[edi],dx

303: 20 62 2c and BYTE PTR [edx+0x2c],ah

306: 20 62 6f and BYTE PTR [edx+0x6f],ah

309: 6f outs dx,DWORD PTR ds:[esi]

30a: 6c ins BYTE PTR es:[edi],dx

30b: 20 63 69 and BYTE PTR [ebx+0x69],ah

30e: 6e outs dx,BYTE PTR ds:[esi]

30f: 2c 20 sub al,0x20

311: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

314: 6c ins BYTE PTR es:[edi],dx

315: 20 26 and BYTE PTR [esi],ah

317: 73 75 jae 0x38e

319: 6d ins DWORD PTR es:[edi],dx

31a: 2c 20 sub al,0x20

31c: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

31f: 6c ins BYTE PTR es:[edi],dx

320: 20 26 and BYTE PTR [esi],ah

322: 63 61 72 arpl WORD PTR [ecx+0x72],sp

325: 72 79 jb 0x3a0

327: 29 20 sub DWORD PTR [eax],esp

329: 7b 0a jnp 0x335

32b: 20 20 and BYTE PTR [eax],ah

32d: 20 20 and BYTE PTR [eax],ah

32f: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

332: 6c ins BYTE PTR es:[edi],dx

333: 20 73 31 and BYTE PTR [ebx+0x31],dh

336: 2c 20 sub al,0x20

338: 63 31 arpl WORD PTR [ecx],si

33a: 2c 20 sub al,0x20

33c: 63 32 arpl WORD PTR [edx],si

33e: 3b 0a cmp ecx,DWORD PTR [edx]

340: 20 20 and BYTE PTR [eax],ah

342: 20 20 and BYTE PTR [eax],ah

344: 68 61 6c 66 41 push 0x41666c61

349: 64 64 65 72 28 fs fs gs jb 0x376

34e: 61 popa

34f: 2c 20 sub al,0x20

351: 62 2c 20 bound ebp,QWORD PTR [eax+eiz\*1]

354: 73 31 jae 0x387

356: 2c 20 sub al,0x20

358: 63 31 arpl WORD PTR [ecx],si

35a: 29 3b sub DWORD PTR [ebx],edi

35c: 0a 20 or ah,BYTE PTR [eax]

35e: 20 20 and BYTE PTR [eax],ah

360: 20 68 61 and BYTE PTR [eax+0x61],ch

363: 6c ins BYTE PTR es:[edi],dx

364: 66 41 inc cx

366: 64 64 65 72 28 fs fs gs jb 0x393

36b: 73 31 jae 0x39e

36d: 2c 20 sub al,0x20

36f: 63 69 6e arpl WORD PTR [ecx+0x6e],bp

372: 2c 20 sub al,0x20

374: 73 75 jae 0x3eb

376: 6d ins DWORD PTR es:[edi],dx

377: 2c 20 sub al,0x20

379: 63 32 arpl WORD PTR [edx],si

37b: 29 3b sub DWORD PTR [ebx],edi

37d: 0a 20 or ah,BYTE PTR [eax]

37f: 20 20 and BYTE PTR [eax],ah

381: 20 63 61 and BYTE PTR [ebx+0x61],ah

384: 72 72 jb 0x3f8

386: 79 20 jns 0x3a8

388: 3d 20 4f 52 28 cmp eax,0x28524f20

38d: 63 31 arpl WORD PTR [ecx],si

38f: 2c 20 sub al,0x20

391: 63 32 arpl WORD PTR [edx],si

393: 29 3b sub DWORD PTR [ebx],edi

395: 0a 7d 0a or bh,BYTE PTR [ebp+0xa]

398: 0a 2f or ch,BYTE PTR [edi]

39a: 2f das

39b: 20 31 and BYTE PTR [ecx],dh

39d: 36 2d 62 69 74 20 ss sub eax,0x20746962

3a3: 41 inc ecx

3a4: 4c dec esp

3a5: 55 push ebp

3a6: 20 28 and BYTE PTR [eax],ch

3a8: 41 inc ecx

3a9: 64 64 2f fs fs das

3ac: 53 push ebx

3ad: 75 62 jne 0x411

3af: 74 72 je 0x423

3b1: 61 popa

3b2: 63 74 2f 4d arpl WORD PTR [edi+ebp\*1+0x4d],si

3b6: 75 6c jne 0x424

3b8: 74 69 je 0x423

3ba: 70 6c jo 0x428

3bc: 79 2f jns 0x3ed

3be: 44 inc esp

3bf: 69 76 69 64 65 29 0a imul esi,DWORD PTR [esi+0x69],0xa296564

3c6: 73 74 jae 0x43c

3c8: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

3cb: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

3ce: 73 65 jae 0x435

3d0: 74 3c je 0x40e

3d2: 31 36 xor DWORD PTR [esi],esi

3d4: 3e 20 41 4c and BYTE PTR ds:[ecx+0x4c],al

3d8: 55 push ebp

3d9: 28 73 74 sub BYTE PTR [ebx+0x74],dh

3dc: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

3df: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

3e2: 73 65 jae 0x449

3e4: 74 3c je 0x422

3e6: 31 36 xor DWORD PTR [esi],esi

3e8: 3e 20 41 2c and BYTE PTR ds:[ecx+0x2c],al

3ec: 20 73 74 and BYTE PTR [ebx+0x74],dh

3ef: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

3f2: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

3f5: 73 65 jae 0x45c

3f7: 74 3c je 0x435

3f9: 31 36 xor DWORD PTR [esi],esi

3fb: 3e 20 42 2c and BYTE PTR ds:[edx+0x2c],al

3ff: 20 62 6f and BYTE PTR [edx+0x6f],ah

402: 6f outs dx,DWORD PTR ds:[esi]

403: 6c ins BYTE PTR es:[edi],dx

404: 20 73 75 and BYTE PTR [ebx+0x75],dh

407: 62 74 72 61 bound esi,QWORD PTR [edx+esi\*2+0x61]

40b: 63 74 2c 20 arpl WORD PTR [esp+ebp\*1+0x20],si

40f: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

412: 6c ins BYTE PTR es:[edi],dx

413: 20 6d 75 and BYTE PTR [ebp+0x75],ch

416: 6c ins BYTE PTR es:[edi],dx

417: 74 69 je 0x482

419: 70 6c jo 0x487

41b: 79 2c jns 0x449

41d: 20 62 6f and BYTE PTR [edx+0x6f],ah

420: 6f outs dx,DWORD PTR ds:[esi]

421: 6c ins BYTE PTR es:[edi],dx

422: 20 64 69 76 and BYTE PTR [ecx+ebp\*2+0x76],ah

426: 69 64 65 29 20 7b 0a imul esp,DWORD PTR [ebp+eiz\*2+0x29],0x200a7b20

42d: 20

42e: 20 20 and BYTE PTR [eax],ah

430: 20 73 74 and BYTE PTR [ebx+0x74],dh

433: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

436: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

439: 73 65 jae 0x4a0

43b: 74 3c je 0x479

43d: 31 36 xor DWORD PTR [esi],esi

43f: 3e 20 72 65 and BYTE PTR ds:[edx+0x65],dh

443: 73 75 jae 0x4ba

445: 6c ins BYTE PTR es:[edi],dx

446: 74 3b je 0x483

448: 0a 20 or ah,BYTE PTR [eax]

44a: 20 20 and BYTE PTR [eax],ah

44c: 20 62 6f and BYTE PTR [edx+0x6f],ah

44f: 6f outs dx,DWORD PTR ds:[esi]

450: 6c ins BYTE PTR es:[edi],dx

451: 20 63 61 and BYTE PTR [ebx+0x61],ah

454: 72 72 jb 0x4c8

456: 79 20 jns 0x478

458: 3d 20 73 75 62 cmp eax,0x62757320

45d: 74 72 je 0x4d1

45f: 61 popa

460: 63 74 3b 0a arpl WORD PTR [ebx+edi\*1+0xa],si

464: 0a 20 or ah,BYTE PTR [eax]

466: 20 20 and BYTE PTR [eax],ah

468: 20 69 66 and BYTE PTR [ecx+0x66],ch

46b: 20 28 and BYTE PTR [eax],ch

46d: 6d ins DWORD PTR es:[edi],dx

46e: 75 6c jne 0x4dc

470: 74 69 je 0x4db

472: 70 6c jo 0x4e0

474: 79 29 jns 0x49f

476: 20 7b 0a and BYTE PTR [ebx+0xa],bh

479: 20 20 and BYTE PTR [eax],ah

47b: 20 20 and BYTE PTR [eax],ah

47d: 20 20 and BYTE PTR [eax],ah

47f: 20 20 and BYTE PTR [eax],ah

481: 72 65 jb 0x4e8

483: 73 75 jae 0x4fa

485: 6c ins BYTE PTR es:[edi],dx

486: 74 20 je 0x4a8

488: 3d 20 73 74 64 cmp eax,0x64747320

48d: 3a 3a cmp bh,BYTE PTR [edx]

48f: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

492: 73 65 jae 0x4f9

494: 74 3c je 0x4d2

496: 31 36 xor DWORD PTR [esi],esi

498: 3e 28 41 2e sub BYTE PTR ds:[ecx+0x2e],al

49c: 74 6f je 0x50d

49e: 5f pop edi

49f: 75 6c jne 0x50d

4a1: 6f outs dx,DWORD PTR ds:[esi]

4a2: 6e outs dx,BYTE PTR ds:[esi]

4a3: 67 28 29 sub BYTE PTR [bx+di],ch

4a6: 20 2a and BYTE PTR [edx],ch

4a8: 20 42 2e and BYTE PTR [edx+0x2e],al

4ab: 74 6f je 0x51c

4ad: 5f pop edi

4ae: 75 6c jne 0x51c

4b0: 6f outs dx,DWORD PTR ds:[esi]

4b1: 6e outs dx,BYTE PTR ds:[esi]

4b2: 67 28 29 sub BYTE PTR [bx+di],ch

4b5: 29 3b sub DWORD PTR [ebx],edi

4b7: 0a 20 or ah,BYTE PTR [eax]

4b9: 20 20 and BYTE PTR [eax],ah

4bb: 20 7d 20 and BYTE PTR [ebp+0x20],bh

4be: 65 6c gs ins BYTE PTR es:[edi],dx

4c0: 73 65 jae 0x527

4c2: 20 69 66 and BYTE PTR [ecx+0x66],ch

4c5: 20 28 and BYTE PTR [eax],ch

4c7: 64 69 76 69 64 65 29 imul esi,DWORD PTR fs:[esi+0x69],0x20296564

4ce: 20

4cf: 7b 0a jnp 0x4db

4d1: 20 20 and BYTE PTR [eax],ah

4d3: 20 20 and BYTE PTR [eax],ah

4d5: 20 20 and BYTE PTR [eax],ah

4d7: 20 20 and BYTE PTR [eax],ah

4d9: 69 66 20 28 42 2e 74 imul esp,DWORD PTR [esi+0x20],0x742e4228

4e0: 6f outs dx,DWORD PTR ds:[esi]

4e1: 5f pop edi

4e2: 75 6c jne 0x550

4e4: 6f outs dx,DWORD PTR ds:[esi]

4e5: 6e outs dx,BYTE PTR ds:[esi]

4e6: 67 28 29 sub BYTE PTR [bx+di],ch

4e9: 20 21 and BYTE PTR [ecx],ah

4eb: 3d 20 30 29 0a cmp eax,0xa293020

4f0: 20 20 and BYTE PTR [eax],ah

4f2: 20 20 and BYTE PTR [eax],ah

4f4: 20 20 and BYTE PTR [eax],ah

4f6: 20 20 and BYTE PTR [eax],ah

4f8: 20 20 and BYTE PTR [eax],ah

4fa: 20 20 and BYTE PTR [eax],ah

4fc: 72 65 jb 0x563

4fe: 73 75 jae 0x575

500: 6c ins BYTE PTR es:[edi],dx

501: 74 20 je 0x523

503: 3d 20 73 74 64 cmp eax,0x64747320

508: 3a 3a cmp bh,BYTE PTR [edx]

50a: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

50d: 73 65 jae 0x574

50f: 74 3c je 0x54d

511: 31 36 xor DWORD PTR [esi],esi

513: 3e 28 41 2e sub BYTE PTR ds:[ecx+0x2e],al

517: 74 6f je 0x588

519: 5f pop edi

51a: 75 6c jne 0x588

51c: 6f outs dx,DWORD PTR ds:[esi]

51d: 6e outs dx,BYTE PTR ds:[esi]

51e: 67 28 29 sub BYTE PTR [bx+di],ch

521: 20 2f and BYTE PTR [edi],ch

523: 20 42 2e and BYTE PTR [edx+0x2e],al

526: 74 6f je 0x597

528: 5f pop edi

529: 75 6c jne 0x597

52b: 6f outs dx,DWORD PTR ds:[esi]

52c: 6e outs dx,BYTE PTR ds:[esi]

52d: 67 28 29 sub BYTE PTR [bx+di],ch

530: 29 3b sub DWORD PTR [ebx],edi

532: 0a 20 or ah,BYTE PTR [eax]

534: 20 20 and BYTE PTR [eax],ah

536: 20 20 and BYTE PTR [eax],ah

538: 20 20 and BYTE PTR [eax],ah

53a: 20 65 6c and BYTE PTR [ebp+0x6c],ah

53d: 73 65 jae 0x5a4

53f: 0a 20 or ah,BYTE PTR [eax]

541: 20 20 and BYTE PTR [eax],ah

543: 20 20 and BYTE PTR [eax],ah

545: 20 20 and BYTE PTR [eax],ah

547: 20 20 and BYTE PTR [eax],ah

549: 20 20 and BYTE PTR [eax],ah

54b: 20 72 65 and BYTE PTR [edx+0x65],dh

54e: 73 75 jae 0x5c5

550: 6c ins BYTE PTR es:[edi],dx

551: 74 20 je 0x573

553: 3d 20 30 3b 20 cmp eax,0x203b3020

558: 2f das

559: 2f das

55a: 20 50 72 and BYTE PTR [eax+0x72],dl

55d: 65 76 65 gs jbe 0x5c5

560: 6e outs dx,BYTE PTR ds:[esi]

561: 74 20 je 0x583

563: 64 69 76 69 73 69 6f imul esi,DWORD PTR fs:[esi+0x69],0x6e6f6973

56a: 6e

56b: 20 62 79 and BYTE PTR [edx+0x79],ah

56e: 20 7a 65 and BYTE PTR [edx+0x65],bh

571: 72 6f jb 0x5e2

573: 0a 20 or ah,BYTE PTR [eax]

575: 20 20 and BYTE PTR [eax],ah

577: 20 7d 20 and BYTE PTR [ebp+0x20],bh

57a: 65 6c gs ins BYTE PTR es:[edi],dx

57c: 73 65 jae 0x5e3

57e: 20 7b 0a and BYTE PTR [ebx+0xa],bh

581: 20 20 and BYTE PTR [eax],ah

583: 20 20 and BYTE PTR [eax],ah

585: 20 20 and BYTE PTR [eax],ah

587: 20 20 and BYTE PTR [eax],ah

589: 66 6f outs dx,WORD PTR ds:[esi]

58b: 72 20 jb 0x5ad

58d: 28 69 6e sub BYTE PTR [ecx+0x6e],ch

590: 74 20 je 0x5b2

592: 69 20 3d 20 30 3b imul esp,DWORD PTR [eax],0x3b30203d

598: 20 69 20 and BYTE PTR [ecx+0x20],ch

59b: 3c 20 cmp al,0x20

59d: 31 36 xor DWORD PTR [esi],esi

59f: 3b 20 cmp esp,DWORD PTR [eax]

5a1: 69 2b 2b 29 20 7b imul ebp,DWORD PTR [ebx],0x7b20292b

5a7: 0a 20 or ah,BYTE PTR [eax]

5a9: 20 20 and BYTE PTR [eax],ah

5ab: 20 20 and BYTE PTR [eax],ah

5ad: 20 20 and BYTE PTR [eax],ah

5af: 20 20 and BYTE PTR [eax],ah

5b1: 20 20 and BYTE PTR [eax],ah

5b3: 20 62 6f and BYTE PTR [edx+0x6f],ah

5b6: 6f outs dx,DWORD PTR ds:[esi]

5b7: 6c ins BYTE PTR es:[edi],dx

5b8: 20 73 75 and BYTE PTR [ebx+0x75],dh

5bb: 6d ins DWORD PTR es:[edi],dx

5bc: 2c 20 sub al,0x20

5be: 63 3b arpl WORD PTR [ebx],di

5c0: 0a 20 or ah,BYTE PTR [eax]

5c2: 20 20 and BYTE PTR [eax],ah

5c4: 20 20 and BYTE PTR [eax],ah

5c6: 20 20 and BYTE PTR [eax],ah

5c8: 20 20 and BYTE PTR [eax],ah

5ca: 20 20 and BYTE PTR [eax],ah

5cc: 20 66 75 and BYTE PTR [esi+0x75],ah

5cf: 6c ins BYTE PTR es:[edi],dx

5d0: 6c ins BYTE PTR es:[edi],dx

5d1: 41 inc ecx

5d2: 64 64 65 72 28 fs fs gs jb 0x5ff

5d7: 41 inc ecx

5d8: 5b pop ebx

5d9: 69 5d 2c 20 58 4f 52 imul ebx,DWORD PTR [ebp+0x2c],0x524f5820

5e0: 28 42 5b sub BYTE PTR [edx+0x5b],al

5e3: 69 5d 2c 20 73 75 62 imul ebx,DWORD PTR [ebp+0x2c],0x62757320

5ea: 74 72 je 0x65e

5ec: 61 popa

5ed: 63 74 29 2c arpl WORD PTR [ecx+ebp\*1+0x2c],si

5f1: 20 63 61 and BYTE PTR [ebx+0x61],ah

5f4: 72 72 jb 0x668

5f6: 79 2c jns 0x624

5f8: 20 73 75 and BYTE PTR [ebx+0x75],dh

5fb: 6d ins DWORD PTR es:[edi],dx

5fc: 2c 20 sub al,0x20

5fe: 63 29 arpl WORD PTR [ecx],bp

600: 3b 0a cmp ecx,DWORD PTR [edx]

602: 20 20 and BYTE PTR [eax],ah

604: 20 20 and BYTE PTR [eax],ah

606: 20 20 and BYTE PTR [eax],ah

608: 20 20 and BYTE PTR [eax],ah

60a: 20 20 and BYTE PTR [eax],ah

60c: 20 20 and BYTE PTR [eax],ah

60e: 72 65 jb 0x675

610: 73 75 jae 0x687

612: 6c ins BYTE PTR es:[edi],dx

613: 74 5b je 0x670

615: 69 5d 20 3d 20 73 75 imul ebx,DWORD PTR [ebp+0x20],0x7573203d

61c: 6d ins DWORD PTR es:[edi],dx

61d: 3b 0a cmp ecx,DWORD PTR [edx]

61f: 20 20 and BYTE PTR [eax],ah

621: 20 20 and BYTE PTR [eax],ah

623: 20 20 and BYTE PTR [eax],ah

625: 20 20 and BYTE PTR [eax],ah

627: 20 20 and BYTE PTR [eax],ah

629: 20 20 and BYTE PTR [eax],ah

62b: 63 61 72 arpl WORD PTR [ecx+0x72],sp

62e: 72 79 jb 0x6a9

630: 20 3d 20 63 3b 0a and BYTE PTR ds:0xa3b6320,bh

636: 20 20 and BYTE PTR [eax],ah

638: 20 20 and BYTE PTR [eax],ah

63a: 20 20 and BYTE PTR [eax],ah

63c: 20 20 and BYTE PTR [eax],ah

63e: 7d 0a jge 0x64a

640: 20 20 and BYTE PTR [eax],ah

642: 20 20 and BYTE PTR [eax],ah

644: 7d 0a jge 0x650

646: 20 20 and BYTE PTR [eax],ah

648: 20 20 and BYTE PTR [eax],ah

64a: 72 65 jb 0x6b1

64c: 74 75 je 0x6c3

64e: 72 6e jb 0x6be

650: 20 72 65 and BYTE PTR [edx+0x65],dh

653: 73 75 jae 0x6ca

655: 6c ins BYTE PTR es:[edi],dx

656: 74 3b je 0x693

658: 0a 7d 0a or bh,BYTE PTR [ebp+0xa]

65b: 0a 2f or ch,BYTE PTR [edi]

65d: 2f das

65e: 20 49 6e and BYTE PTR [ecx+0x6e],cl

661: 63 72 65 arpl WORD PTR [edx+0x65],si

664: 6d ins DWORD PTR es:[edi],dx

665: 65 6e outs dx,BYTE PTR gs:[esi]

667: 74 20 je 0x689

669: 26 20 44 65 63 and BYTE PTR es:[ebp+eiz\*2+0x63],al

66e: 72 65 jb 0x6d5

670: 6d ins DWORD PTR es:[edi],dx

671: 65 6e outs dx,BYTE PTR gs:[esi]

673: 74 0a je 0x67f

675: 73 74 jae 0x6eb

677: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

67a: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

67d: 73 65 jae 0x6e4

67f: 74 3c je 0x6bd

681: 31 36 xor DWORD PTR [esi],esi

683: 3e 20 49 4e and BYTE PTR ds:[ecx+0x4e],cl

687: 43 inc ebx

688: 28 73 74 sub BYTE PTR [ebx+0x74],dh

68b: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

68e: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

691: 73 65 jae 0x6f8

693: 74 3c je 0x6d1

695: 31 36 xor DWORD PTR [esi],esi

697: 3e 20 41 29 and BYTE PTR ds:[ecx+0x29],al

69b: 20 7b 0a and BYTE PTR [ebx+0xa],bh

69e: 20 20 and BYTE PTR [eax],ah

6a0: 20 20 and BYTE PTR [eax],ah

6a2: 72 65 jb 0x709

6a4: 74 75 je 0x71b

6a6: 72 6e jb 0x716

6a8: 20 41 4c and BYTE PTR [ecx+0x4c],al

6ab: 55 push ebp

6ac: 28 41 2c sub BYTE PTR [ecx+0x2c],al

6af: 20 73 74 and BYTE PTR [ebx+0x74],dh

6b2: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

6b5: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

6b8: 73 65 jae 0x71f

6ba: 74 3c je 0x6f8

6bc: 31 36 xor DWORD PTR [esi],esi

6be: 3e 28 31 sub BYTE PTR ds:[ecx],dh

6c1: 29 2c 20 sub DWORD PTR [eax+eiz\*1],ebp

6c4: 66 61 popaw

6c6: 6c ins BYTE PTR es:[edi],dx

6c7: 73 65 jae 0x72e

6c9: 2c 20 sub al,0x20

6cb: 66 61 popaw

6cd: 6c ins BYTE PTR es:[edi],dx

6ce: 73 65 jae 0x735

6d0: 2c 20 sub al,0x20

6d2: 66 61 popaw

6d4: 6c ins BYTE PTR es:[edi],dx

6d5: 73 65 jae 0x73c

6d7: 29 3b sub DWORD PTR [ebx],edi

6d9: 0a 7d 0a or bh,BYTE PTR [ebp+0xa]

6dc: 73 74 jae 0x752

6de: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

6e1: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

6e4: 73 65 jae 0x74b

6e6: 74 3c je 0x724

6e8: 31 36 xor DWORD PTR [esi],esi

6ea: 3e 20 44 45 43 and BYTE PTR ds:[ebp+eax\*2+0x43],al

6ef: 28 73 74 sub BYTE PTR [ebx+0x74],dh

6f2: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

6f5: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

6f8: 73 65 jae 0x75f

6fa: 74 3c je 0x738

6fc: 31 36 xor DWORD PTR [esi],esi

6fe: 3e 20 41 29 and BYTE PTR ds:[ecx+0x29],al

702: 20 7b 0a and BYTE PTR [ebx+0xa],bh

705: 20 20 and BYTE PTR [eax],ah

707: 20 20 and BYTE PTR [eax],ah

709: 72 65 jb 0x770

70b: 74 75 je 0x782

70d: 72 6e jb 0x77d

70f: 20 41 4c and BYTE PTR [ecx+0x4c],al

712: 55 push ebp

713: 28 41 2c sub BYTE PTR [ecx+0x2c],al

716: 20 73 74 and BYTE PTR [ebx+0x74],dh

719: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

71c: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

71f: 73 65 jae 0x786

721: 74 3c je 0x75f

723: 31 36 xor DWORD PTR [esi],esi

725: 3e 28 31 sub BYTE PTR ds:[ecx],dh

728: 29 2c 20 sub DWORD PTR [eax+eiz\*1],ebp

72b: 74 72 je 0x79f

72d: 75 65 jne 0x794

72f: 2c 20 sub al,0x20

731: 66 61 popaw

733: 6c ins BYTE PTR es:[edi],dx

734: 73 65 jae 0x79b

736: 2c 20 sub al,0x20

738: 66 61 popaw

73a: 6c ins BYTE PTR es:[edi],dx

73b: 73 65 jae 0x7a2

73d: 29 3b sub DWORD PTR [ebx],edi

73f: 0a 7d 0a or bh,BYTE PTR [ebp+0xa]

742: 0a 2f or ch,BYTE PTR [edi]

744: 2f das

745: 20 53 68 and BYTE PTR [ebx+0x68],dl

748: 69 66 74 20 4c 65 66 imul esp,DWORD PTR [esi+0x74],0x66654c20

74f: 74 20 je 0x771

751: 61 popa

752: 6e outs dx,BYTE PTR ds:[esi]

753: 64 20 53 68 and BYTE PTR fs:[ebx+0x68],dl

757: 69 66 74 20 52 69 67 imul esp,DWORD PTR [esi+0x74],0x67695220

75e: 68 74 0a 73 74 push 0x74730a74

763: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

766: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

769: 73 65 jae 0x7d0

76b: 74 3c je 0x7a9

76d: 31 36 xor DWORD PTR [esi],esi

76f: 3e 20 53 48 and BYTE PTR ds:[ebx+0x48],dl

773: 4c dec esp

774: 28 73 74 sub BYTE PTR [ebx+0x74],dh

777: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

77a: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

77d: 73 65 jae 0x7e4

77f: 74 3c je 0x7bd

781: 31 36 xor DWORD PTR [esi],esi

783: 3e 20 41 29 and BYTE PTR ds:[ecx+0x29],al

787: 20 7b 0a and BYTE PTR [ebx+0xa],bh

78a: 20 20 and BYTE PTR [eax],ah

78c: 20 20 and BYTE PTR [eax],ah

78e: 72 65 jb 0x7f5

790: 74 75 je 0x807

792: 72 6e jb 0x802

794: 20 41 20 and BYTE PTR [ecx+0x20],al

797: 3c 3c cmp al,0x3c

799: 20 31 and BYTE PTR [ecx],dh

79b: 3b 0a cmp ecx,DWORD PTR [edx]

79d: 7d 0a jge 0x7a9

79f: 0a 73 74 or dh,BYTE PTR [ebx+0x74]

7a2: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

7a5: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

7a8: 73 65 jae 0x80f

7aa: 74 3c je 0x7e8

7ac: 31 36 xor DWORD PTR [esi],esi

7ae: 3e 20 53 48 and BYTE PTR ds:[ebx+0x48],dl

7b2: 52 push edx

7b3: 28 73 74 sub BYTE PTR [ebx+0x74],dh

7b6: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

7b9: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

7bc: 73 65 jae 0x823

7be: 74 3c je 0x7fc

7c0: 31 36 xor DWORD PTR [esi],esi

7c2: 3e 20 41 29 and BYTE PTR ds:[ecx+0x29],al

7c6: 20 7b 0a and BYTE PTR [ebx+0xa],bh

7c9: 20 20 and BYTE PTR [eax],ah

7cb: 20 20 and BYTE PTR [eax],ah

7cd: 72 65 jb 0x834

7cf: 74 75 je 0x846

7d1: 72 6e jb 0x841

7d3: 20 41 20 and BYTE PTR [ecx+0x20],al

7d6: 3e 3e 20 31 ds and BYTE PTR ds:[ecx],dh

7da: 3b 0a cmp ecx,DWORD PTR [edx]

7dc: 7d 0a jge 0x7e8

7de: 0a 2f or ch,BYTE PTR [edi]

7e0: 2f das

7e1: 20 52 65 and BYTE PTR [edx+0x65],dl

7e4: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x20737265

7eb: 20

7ec: 61 popa

7ed: 6e outs dx,BYTE PTR ds:[esi]

7ee: 64 20 4d 65 and BYTE PTR fs:[ebp+0x65],cl

7f2: 6d ins DWORD PTR es:[edi],dx

7f3: 6f outs dx,DWORD PTR ds:[esi]

7f4: 72 79 jb 0x86f

7f6: 0a 73 74 or dh,BYTE PTR [ebx+0x74]

7f9: 72 75 jb 0x870

7fb: 63 74 20 43 arpl WORD PTR [eax+eiz\*1+0x43],si

7ff: 50 push eax

800: 55 push ebp

801: 20 7b 0a and BYTE PTR [ebx+0xa],bh

804: 20 20 and BYTE PTR [eax],ah

806: 20 20 and BYTE PTR [eax],ah

808: 73 74 jae 0x87e

80a: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

80d: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

810: 73 65 jae 0x877

812: 74 3c je 0x850

814: 31 36 xor DWORD PTR [esi],esi

816: 3e 20 72 65 and BYTE PTR ds:[edx+0x65],dh

81a: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

821: 5b

822: 38 5d 3b cmp BYTE PTR [ebp+0x3b],bl

825: 20 2f and BYTE PTR [edi],ch

827: 2f das

828: 20 38 and BYTE PTR [eax],bh

82a: 20 67 65 and BYTE PTR [edi+0x65],ah

82d: 6e outs dx,BYTE PTR ds:[esi]

82e: 65 72 61 gs jb 0x892

831: 6c ins BYTE PTR es:[edi],dx

832: 2d 70 75 72 70 sub eax,0x70727570

837: 6f outs dx,DWORD PTR ds:[esi]

838: 73 65 jae 0x89f

83a: 20 72 65 and BYTE PTR [edx+0x65],dh

83d: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0xa737265

844: 0a

845: 20 20 and BYTE PTR [eax],ah

847: 20 20 and BYTE PTR [eax],ah

849: 73 74 jae 0x8bf

84b: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

84e: 76 65 jbe 0x8b5

850: 63 74 6f 72 arpl WORD PTR [edi+ebp\*2+0x72],si

854: 3c 73 cmp al,0x73

856: 74 64 je 0x8bc

858: 3a 3a cmp bh,BYTE PTR [edx]

85a: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

85d: 73 65 jae 0x8c4

85f: 74 3c je 0x89d

861: 31 36 xor DWORD PTR [esi],esi

863: 3e 3e 20 6d 65 ds and BYTE PTR ds:[ebp+0x65],ch

868: 6d ins DWORD PTR es:[edi],dx

869: 6f outs dx,DWORD PTR ds:[esi]

86a: 72 79 jb 0x8e5

86c: 3b 0a cmp ecx,DWORD PTR [edx]

86e: 0a 20 or ah,BYTE PTR [eax]

870: 20 20 and BYTE PTR [eax],ah

872: 20 2f and BYTE PTR [edi],ch

874: 2f das

875: 20 53 74 and BYTE PTR [ebx+0x74],dl

878: 61 popa

879: 63 6b 20 arpl WORD PTR [ebx+0x20],bp

87c: 70 6f jo 0x8ed

87e: 69 6e 74 65 72 20 61 imul ebp,DWORD PTR [esi+0x74],0x61207265

885: 6e outs dx,BYTE PTR ds:[esi]

886: 64 20 66 6c and BYTE PTR fs:[esi+0x6c],ah

88a: 61 popa

88b: 67 73 0a addr16 jae 0x898

88e: 20 20 and BYTE PTR [eax],ah

890: 20 20 and BYTE PTR [eax],ah

892: 69 6e 74 20 73 70 20 imul ebp,DWORD PTR [esi+0x74],0x20707320

899: 3d 20 30 3b 20 cmp eax,0x203b3020

89e: 2f das

89f: 2f das

8a0: 20 53 74 and BYTE PTR [ebx+0x74],dl

8a3: 61 popa

8a4: 63 6b 20 arpl WORD PTR [ebx+0x20],bp

8a7: 70 6f jo 0x918

8a9: 69 6e 74 65 72 20 66 imul ebp,DWORD PTR [esi+0x74],0x66207265

8b0: 6f outs dx,DWORD PTR ds:[esi]

8b1: 72 20 jb 0x8d3

8b3: 50 push eax

8b4: 55 push ebp

8b5: 53 push ebx

8b6: 48 dec eax

8b7: 2f das

8b8: 50 push eax

8b9: 4f dec edi

8ba: 50 push eax

8bb: 20 6f 70 and BYTE PTR [edi+0x70],ch

8be: 65 72 61 gs jb 0x922

8c1: 74 69 je 0x92c

8c3: 6f outs dx,DWORD PTR ds:[esi]

8c4: 6e outs dx,BYTE PTR ds:[esi]

8c5: 73 0a jae 0x8d1

8c7: 20 20 and BYTE PTR [eax],ah

8c9: 20 20 and BYTE PTR [eax],ah

8cb: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

8ce: 6c ins BYTE PTR es:[edi],dx

8cf: 20 7a 65 and BYTE PTR [edx+0x65],bh

8d2: 72 6f jb 0x943

8d4: 46 inc esi

8d5: 6c ins BYTE PTR es:[edi],dx

8d6: 61 popa

8d7: 67 20 3d and BYTE PTR [di],bh

8da: 20 66 61 and BYTE PTR [esi+0x61],ah

8dd: 6c ins BYTE PTR es:[edi],dx

8de: 73 65 jae 0x945

8e0: 3b 0a cmp ecx,DWORD PTR [edx]

8e2: 20 20 and BYTE PTR [eax],ah

8e4: 20 20 and BYTE PTR [eax],ah

8e6: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

8e9: 6c ins BYTE PTR es:[edi],dx

8ea: 20 63 61 and BYTE PTR [ebx+0x61],ah

8ed: 72 72 jb 0x961

8ef: 79 46 jns 0x937

8f1: 6c ins BYTE PTR es:[edi],dx

8f2: 61 popa

8f3: 67 20 3d and BYTE PTR [di],bh

8f6: 20 66 61 and BYTE PTR [esi+0x61],ah

8f9: 6c ins BYTE PTR es:[edi],dx

8fa: 73 65 jae 0x961

8fc: 3b 0a cmp ecx,DWORD PTR [edx]

8fe: 0a 20 or ah,BYTE PTR [eax]

900: 20 20 and BYTE PTR [eax],ah

902: 20 43 50 and BYTE PTR [ebx+0x50],al

905: 55 push ebp

906: 28 73 69 sub BYTE PTR [ebx+0x69],dh

909: 7a 65 jp 0x970

90b: 5f pop edi

90c: 74 20 je 0x92e

90e: 6d ins DWORD PTR es:[edi],dx

90f: 65 6d gs ins DWORD PTR es:[edi],dx

911: 6f outs dx,DWORD PTR ds:[esi]

912: 72 79 jb 0x98d

914: 5f pop edi

915: 73 69 jae 0x980

917: 7a 65 jp 0x97e

919: 29 20 sub DWORD PTR [eax],esp

91b: 3a 20 cmp ah,BYTE PTR [eax]

91d: 6d ins DWORD PTR es:[edi],dx

91e: 65 6d gs ins DWORD PTR es:[edi],dx

920: 6f outs dx,DWORD PTR ds:[esi]

921: 72 79 jb 0x99c

923: 28 6d 65 sub BYTE PTR [ebp+0x65],ch

926: 6d ins DWORD PTR es:[edi],dx

927: 6f outs dx,DWORD PTR ds:[esi]

928: 72 79 jb 0x9a3

92a: 5f pop edi

92b: 73 69 jae 0x996

92d: 7a 65 jp 0x994

92f: 29 20 sub DWORD PTR [eax],esp

931: 7b 7d jnp 0x9b0

933: 0a 0a or cl,BYTE PTR [edx]

935: 20 20 and BYTE PTR [eax],ah

937: 20 20 and BYTE PTR [eax],ah

939: 2f das

93a: 2f das

93b: 20 4d 4f and BYTE PTR [ebp+0x4f],cl

93e: 56 push esi

93f: 20 69 6e and BYTE PTR [ecx+0x6e],ch

942: 73 74 jae 0x9b8

944: 72 75 jb 0x9bb

946: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

94a: 6e outs dx,BYTE PTR ds:[esi]

94b: 3a 20 cmp ah,BYTE PTR [eax]

94d: 4d dec ebp

94e: 6f outs dx,DWORD PTR ds:[esi]

94f: 76 65 jbe 0x9b6

951: 20 76 61 and BYTE PTR [esi+0x61],dh

954: 6c ins BYTE PTR es:[edi],dx

955: 75 65 jne 0x9bc

957: 20 69 6e and BYTE PTR [ecx+0x6e],ch

95a: 74 6f je 0x9cb

95c: 20 72 65 and BYTE PTR [edx+0x65],dh

95f: 67 69 73 74 65 72 0a imul esi,DWORD PTR [bp+di+0x74],0x200a7265

966: 20

967: 20 20 and BYTE PTR [eax],ah

969: 20 76 6f and BYTE PTR [esi+0x6f],dh

96c: 69 64 20 4d 4f 56 28 imul esp,DWORD PTR [eax+eiz\*1+0x4d],0x6928564f

973: 69

974: 6e outs dx,BYTE PTR ds:[esi]

975: 74 20 je 0x997

977: 72 65 jb 0x9de

979: 67 2c 20 addr16 sub al,0x20

97c: 73 74 jae 0x9f2

97e: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

981: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

984: 73 65 jae 0x9eb

986: 74 3c je 0x9c4

988: 31 36 xor DWORD PTR [esi],esi

98a: 3e 20 76 61 and BYTE PTR ds:[esi+0x61],dh

98e: 6c ins BYTE PTR es:[edi],dx

98f: 75 65 jne 0x9f6

991: 29 20 sub DWORD PTR [eax],esp

993: 7b 0a jnp 0x99f

995: 20 20 and BYTE PTR [eax],ah

997: 20 20 and BYTE PTR [eax],ah

999: 20 20 and BYTE PTR [eax],ah

99b: 20 20 and BYTE PTR [eax],ah

99d: 72 65 jb 0xa04

99f: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

9a6: 5b

9a7: 72 65 jb 0xa0e

9a9: 67 5d addr16 pop ebp

9ab: 20 3d 20 76 61 6c and BYTE PTR ds:0x6c617620,bh

9b1: 75 65 jne 0xa18

9b3: 3b 0a cmp ecx,DWORD PTR [edx]

9b5: 20 20 and BYTE PTR [eax],ah

9b7: 20 20 and BYTE PTR [eax],ah

9b9: 7d 0a jge 0x9c5

9bb: 0a 20 or ah,BYTE PTR [eax]

9bd: 20 20 and BYTE PTR [eax],ah

9bf: 20 2f and BYTE PTR [edi],ch

9c1: 2f das

9c2: 20 4a 4d and BYTE PTR [edx+0x4d],cl

9c5: 50 push eax

9c6: 20 69 6e and BYTE PTR [ecx+0x6e],ch

9c9: 73 74 jae 0xa3f

9cb: 72 75 jb 0xa42

9cd: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

9d1: 6e outs dx,BYTE PTR ds:[esi]

9d2: 3a 20 cmp ah,BYTE PTR [eax]

9d4: 4a dec edx

9d5: 75 6d jne 0xa44

9d7: 70 20 jo 0x9f9

9d9: 74 6f je 0xa4a

9db: 20 73 70 and BYTE PTR [ebx+0x70],dh

9de: 65 63 69 66 arpl WORD PTR gs:[ecx+0x66],bp

9e2: 69 63 20 6d 65 6d 6f imul esp,DWORD PTR [ebx+0x20],0x6f6d656d

9e9: 72 79 jb 0xa64

9eb: 20 61 64 and BYTE PTR [ecx+0x64],ah

9ee: 64 72 65 fs jb 0xa56

9f1: 73 73 jae 0xa66

9f3: 20 28 and BYTE PTR [eax],ch

9f5: 75 6e jne 0xa65

9f7: 63 6f 6e arpl WORD PTR [edi+0x6e],bp

9fa: 64 69 74 69 6f 6e 61 imul esi,DWORD PTR fs:[ecx+ebp\*2+0x6f],0x296c616e

a01: 6c 29

a03: 0a 20 or ah,BYTE PTR [eax]

a05: 20 20 and BYTE PTR [eax],ah

a07: 20 76 6f and BYTE PTR [esi+0x6f],dh

a0a: 69 64 20 4a 4d 50 28 imul esp,DWORD PTR [eax+eiz\*1+0x4a],0x6928504d

a11: 69

a12: 6e outs dx,BYTE PTR ds:[esi]

a13: 74 20 je 0xa35

a15: 61 popa

a16: 64 64 72 65 fs fs jb 0xa7f

a1a: 73 73 jae 0xa8f

a1c: 29 20 sub DWORD PTR [eax],esp

a1e: 7b 0a jnp 0xa2a

a20: 20 20 and BYTE PTR [eax],ah

a22: 20 20 and BYTE PTR [eax],ah

a24: 20 20 and BYTE PTR [eax],ah

a26: 20 20 and BYTE PTR [eax],ah

a28: 73 70 jae 0xa9a

a2a: 20 3d 20 61 64 64 and BYTE PTR ds:0x64646120,bh

a30: 72 65 jb 0xa97

a32: 73 73 jae 0xaa7

a34: 3b 20 cmp esp,DWORD PTR [eax]

a36: 2f das

a37: 2f das

a38: 20 53 65 and BYTE PTR [ebx+0x65],dl

a3b: 74 20 je 0xa5d

a3d: 73 74 jae 0xab3

a3f: 61 popa

a40: 63 6b 20 arpl WORD PTR [ebx+0x20],bp

a43: 70 6f jo 0xab4

a45: 69 6e 74 65 72 20 74 imul ebp,DWORD PTR [esi+0x74],0x74207265

a4c: 6f outs dx,DWORD PTR ds:[esi]

a4d: 20 74 68 65 and BYTE PTR [eax+ebp\*2+0x65],dh

a51: 20 6a 75 and BYTE PTR [edx+0x75],ch

a54: 6d ins DWORD PTR es:[edi],dx

a55: 70 20 jo 0xa77

a57: 61 popa

a58: 64 64 72 65 fs fs jb 0xac1

a5c: 73 73 jae 0xad1

a5e: 20 28 and BYTE PTR [eax],ch

a60: 73 69 jae 0xacb

a62: 6d ins DWORD PTR es:[edi],dx

a63: 75 6c jne 0xad1

a65: 61 popa

a66: 74 69 je 0xad1

a68: 6e outs dx,BYTE PTR ds:[esi]

a69: 67 20 6a 75 and BYTE PTR [bp+si+0x75],ch

a6d: 6d ins DWORD PTR es:[edi],dx

a6e: 70 29 jo 0xa99

a70: 0a 20 or ah,BYTE PTR [eax]

a72: 20 20 and BYTE PTR [eax],ah

a74: 20 7d 0a and BYTE PTR [ebp+0xa],bh

a77: 0a 20 or ah,BYTE PTR [eax]

a79: 20 20 and BYTE PTR [eax],ah

a7b: 20 2f and BYTE PTR [edi],ch

a7d: 2f das

a7e: 20 4a 43 and BYTE PTR [edx+0x43],cl

a81: 20 69 6e and BYTE PTR [ecx+0x6e],ch

a84: 73 74 jae 0xafa

a86: 72 75 jb 0xafd

a88: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

a8c: 6e outs dx,BYTE PTR ds:[esi]

a8d: 3a 20 cmp ah,BYTE PTR [eax]

a8f: 4a dec edx

a90: 75 6d jne 0xaff

a92: 70 20 jo 0xab4

a94: 69 66 20 63 61 72 72 imul esp,DWORD PTR [esi+0x20],0x72726163

a9b: 79 20 jns 0xabd

a9d: 66 6c data16 ins BYTE PTR es:[edi],dx

a9f: 61 popa

aa0: 67 20 69 73 and BYTE PTR [bx+di+0x73],ch

aa4: 20 73 65 and BYTE PTR [ebx+0x65],dh

aa7: 74 0a je 0xab3

aa9: 20 20 and BYTE PTR [eax],ah

aab: 20 20 and BYTE PTR [eax],ah

aad: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

ab0: 6c ins BYTE PTR es:[edi],dx

ab1: 20 4a 43 and BYTE PTR [edx+0x43],cl

ab4: 28 29 sub BYTE PTR [ecx],ch

ab6: 20 7b 0a and BYTE PTR [ebx+0xa],bh

ab9: 20 20 and BYTE PTR [eax],ah

abb: 20 20 and BYTE PTR [eax],ah

abd: 20 20 and BYTE PTR [eax],ah

abf: 20 20 and BYTE PTR [eax],ah

ac1: 72 65 jb 0xb28

ac3: 74 75 je 0xb3a

ac5: 72 6e jb 0xb35

ac7: 20 63 61 and BYTE PTR [ebx+0x61],ah

aca: 72 72 jb 0xb3e

acc: 79 46 jns 0xb14

ace: 6c ins BYTE PTR es:[edi],dx

acf: 61 popa

ad0: 67 3b 20 cmp esp,DWORD PTR [bx+si]

ad3: 2f das

ad4: 2f das

ad5: 20 4a 75 and BYTE PTR [edx+0x75],cl

ad8: 6d ins DWORD PTR es:[edi],dx

ad9: 70 20 jo 0xafb

adb: 69 66 20 63 61 72 72 imul esp,DWORD PTR [esi+0x20],0x72726163

ae2: 79 20 jns 0xb04

ae4: 66 6c data16 ins BYTE PTR es:[edi],dx

ae6: 61 popa

ae7: 67 20 69 73 and BYTE PTR [bx+di+0x73],ch

aeb: 20 73 65 and BYTE PTR [ebx+0x65],dh

aee: 74 0a je 0xafa

af0: 20 20 and BYTE PTR [eax],ah

af2: 20 20 and BYTE PTR [eax],ah

af4: 7d 0a jge 0xb00

af6: 0a 20 or ah,BYTE PTR [eax]

af8: 20 20 and BYTE PTR [eax],ah

afa: 20 2f and BYTE PTR [edi],ch

afc: 2f das

afd: 20 4a 5a and BYTE PTR [edx+0x5a],cl

b00: 20 69 6e and BYTE PTR [ecx+0x6e],ch

b03: 73 74 jae 0xb79

b05: 72 75 jb 0xb7c

b07: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

b0b: 6e outs dx,BYTE PTR ds:[esi]

b0c: 3a 20 cmp ah,BYTE PTR [eax]

b0e: 4a dec edx

b0f: 75 6d jne 0xb7e

b11: 70 20 jo 0xb33

b13: 69 66 20 7a 65 72 6f imul esp,DWORD PTR [esi+0x20],0x6f72657a

b1a: 20 66 6c and BYTE PTR [esi+0x6c],ah

b1d: 61 popa

b1e: 67 20 69 73 and BYTE PTR [bx+di+0x73],ch

b22: 20 73 65 and BYTE PTR [ebx+0x65],dh

b25: 74 0a je 0xb31

b27: 20 20 and BYTE PTR [eax],ah

b29: 20 20 and BYTE PTR [eax],ah

b2b: 62 6f 6f bound ebp,QWORD PTR [edi+0x6f]

b2e: 6c ins BYTE PTR es:[edi],dx

b2f: 20 4a 5a and BYTE PTR [edx+0x5a],cl

b32: 28 29 sub BYTE PTR [ecx],ch

b34: 20 7b 0a and BYTE PTR [ebx+0xa],bh

b37: 20 20 and BYTE PTR [eax],ah

b39: 20 20 and BYTE PTR [eax],ah

b3b: 20 20 and BYTE PTR [eax],ah

b3d: 20 20 and BYTE PTR [eax],ah

b3f: 72 65 jb 0xba6

b41: 74 75 je 0xbb8

b43: 72 6e jb 0xbb3

b45: 20 7a 65 and BYTE PTR [edx+0x65],bh

b48: 72 6f jb 0xbb9

b4a: 46 inc esi

b4b: 6c ins BYTE PTR es:[edi],dx

b4c: 61 popa

b4d: 67 3b 20 cmp esp,DWORD PTR [bx+si]

b50: 2f das

b51: 2f das

b52: 20 4a 75 and BYTE PTR [edx+0x75],cl

b55: 6d ins DWORD PTR es:[edi],dx

b56: 70 20 jo 0xb78

b58: 69 66 20 7a 65 72 6f imul esp,DWORD PTR [esi+0x20],0x6f72657a

b5f: 20 66 6c and BYTE PTR [esi+0x6c],ah

b62: 61 popa

b63: 67 20 69 73 and BYTE PTR [bx+di+0x73],ch

b67: 20 73 65 and BYTE PTR [ebx+0x65],dh

b6a: 74 0a je 0xb76

b6c: 20 20 and BYTE PTR [eax],ah

b6e: 20 20 and BYTE PTR [eax],ah

b70: 7d 0a jge 0xb7c

b72: 0a 20 or ah,BYTE PTR [eax]

b74: 20 20 and BYTE PTR [eax],ah

b76: 20 2f and BYTE PTR [edi],ch

b78: 2f das

b79: 20 50 55 and BYTE PTR [eax+0x55],dl

b7c: 53 push ebx

b7d: 48 dec eax

b7e: 20 69 6e and BYTE PTR [ecx+0x6e],ch

b81: 73 74 jae 0xbf7

b83: 72 75 jb 0xbfa

b85: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

b89: 6e outs dx,BYTE PTR ds:[esi]

b8a: 3a 20 cmp ah,BYTE PTR [eax]

b8c: 50 push eax

b8d: 75 73 jne 0xc02

b8f: 68 20 72 65 67 push 0x67657220

b94: 69 73 74 65 72 20 76 imul esi,DWORD PTR [ebx+0x74],0x76207265

b9b: 61 popa

b9c: 6c ins BYTE PTR es:[edi],dx

b9d: 75 65 jne 0xc04

b9f: 20 74 6f 20 and BYTE PTR [edi+ebp\*2+0x20],dh

ba3: 73 74 jae 0xc19

ba5: 61 popa

ba6: 63 6b 0a arpl WORD PTR [ebx+0xa],bp

ba9: 20 20 and BYTE PTR [eax],ah

bab: 20 20 and BYTE PTR [eax],ah

bad: 76 6f jbe 0xc1e

baf: 69 64 20 50 55 53 48 imul esp,DWORD PTR [eax+eiz\*1+0x50],0x28485355

bb6: 28

bb7: 69 6e 74 20 72 65 67 imul ebp,DWORD PTR [esi+0x74],0x67657220

bbe: 29 20 sub DWORD PTR [eax],esp

bc0: 7b 0a jnp 0xbcc

bc2: 20 20 and BYTE PTR [eax],ah

bc4: 20 20 and BYTE PTR [eax],ah

bc6: 20 20 and BYTE PTR [eax],ah

bc8: 20 20 and BYTE PTR [eax],ah

bca: 69 66 20 28 73 70 20 imul esp,DWORD PTR [esi+0x20],0x20707328

bd1: 3c 20 cmp al,0x20

bd3: 6d ins DWORD PTR es:[edi],dx

bd4: 65 6d gs ins DWORD PTR es:[edi],dx

bd6: 6f outs dx,DWORD PTR ds:[esi]

bd7: 72 79 jb 0xc52

bd9: 2e 73 69 cs jae 0xc45

bdc: 7a 65 jp 0xc43

bde: 28 29 sub BYTE PTR [ecx],ch

be0: 29 20 sub DWORD PTR [eax],esp

be2: 7b 0a jnp 0xbee

be4: 20 20 and BYTE PTR [eax],ah

be6: 20 20 and BYTE PTR [eax],ah

be8: 20 20 and BYTE PTR [eax],ah

bea: 20 20 and BYTE PTR [eax],ah

bec: 20 20 and BYTE PTR [eax],ah

bee: 20 20 and BYTE PTR [eax],ah

bf0: 6d ins DWORD PTR es:[edi],dx

bf1: 65 6d gs ins DWORD PTR es:[edi],dx

bf3: 6f outs dx,DWORD PTR ds:[esi]

bf4: 72 79 jb 0xc6f

bf6: 5b pop ebx

bf7: 73 70 jae 0xc69

bf9: 5d pop ebp

bfa: 20 3d 20 72 65 67 and BYTE PTR ds:0x67657220,bh

c00: 69 73 74 65 72 73 5b imul esi,DWORD PTR [ebx+0x74],0x5b737265

c07: 72 65 jb 0xc6e

c09: 67 5d addr16 pop ebp

c0b: 3b 20 cmp esp,DWORD PTR [eax]

c0d: 2f das

c0e: 2f das

c0f: 20 53 74 and BYTE PTR [ebx+0x74],dl

c12: 6f outs dx,DWORD PTR ds:[esi]

c13: 72 65 jb 0xc7a

c15: 20 76 61 and BYTE PTR [esi+0x61],dh

c18: 6c ins BYTE PTR es:[edi],dx

c19: 75 65 jne 0xc80

c1b: 20 69 6e and BYTE PTR [ecx+0x6e],ch

c1e: 20 6d 65 and BYTE PTR [ebp+0x65],ch

c21: 6d ins DWORD PTR es:[edi],dx

c22: 6f outs dx,DWORD PTR ds:[esi]

c23: 72 79 jb 0xc9e

c25: 0a 20 or ah,BYTE PTR [eax]

c27: 20 20 and BYTE PTR [eax],ah

c29: 20 20 and BYTE PTR [eax],ah

c2b: 20 20 and BYTE PTR [eax],ah

c2d: 20 20 and BYTE PTR [eax],ah

c2f: 20 20 and BYTE PTR [eax],ah

c31: 20 73 70 and BYTE PTR [ebx+0x70],dh

c34: 2b 2b sub ebp,DWORD PTR [ebx]

c36: 3b 0a cmp ecx,DWORD PTR [edx]

c38: 20 20 and BYTE PTR [eax],ah

c3a: 20 20 and BYTE PTR [eax],ah

c3c: 20 20 and BYTE PTR [eax],ah

c3e: 20 20 and BYTE PTR [eax],ah

c40: 7d 0a jge 0xc4c

c42: 20 20 and BYTE PTR [eax],ah

c44: 20 20 and BYTE PTR [eax],ah

c46: 7d 0a jge 0xc52

c48: 0a 20 or ah,BYTE PTR [eax]

c4a: 20 20 and BYTE PTR [eax],ah

c4c: 20 2f and BYTE PTR [edi],ch

c4e: 2f das

c4f: 20 50 4f and BYTE PTR [eax+0x4f],dl

c52: 50 push eax

c53: 20 69 6e and BYTE PTR [ecx+0x6e],ch

c56: 73 74 jae 0xccc

c58: 72 75 jb 0xccf

c5a: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

c5e: 6e outs dx,BYTE PTR ds:[esi]

c5f: 3a 20 cmp ah,BYTE PTR [eax]

c61: 50 push eax

c62: 6f outs dx,DWORD PTR ds:[esi]

c63: 70 20 jo 0xc85

c65: 76 61 jbe 0xcc8

c67: 6c ins BYTE PTR es:[edi],dx

c68: 75 65 jne 0xccf

c6a: 20 66 72 and BYTE PTR [esi+0x72],ah

c6d: 6f outs dx,DWORD PTR ds:[esi]

c6e: 6d ins DWORD PTR es:[edi],dx

c6f: 20 73 74 and BYTE PTR [ebx+0x74],dh

c72: 61 popa

c73: 63 6b 20 arpl WORD PTR [ebx+0x20],bp

c76: 69 6e 74 6f 20 72 65 imul ebp,DWORD PTR [esi+0x74],0x6572206f

c7d: 67 69 73 74 65 72 0a imul esi,DWORD PTR [bp+di+0x74],0x200a7265

c84: 20

c85: 20 20 and BYTE PTR [eax],ah

c87: 20 76 6f and BYTE PTR [esi+0x6f],dh

c8a: 69 64 20 50 4f 50 28 imul esp,DWORD PTR [eax+eiz\*1+0x50],0x6928504f

c91: 69

c92: 6e outs dx,BYTE PTR ds:[esi]

c93: 74 20 je 0xcb5

c95: 72 65 jb 0xcfc

c97: 67 29 20 sub DWORD PTR [bx+si],esp

c9a: 7b 0a jnp 0xca6

c9c: 20 20 and BYTE PTR [eax],ah

c9e: 20 20 and BYTE PTR [eax],ah

ca0: 20 20 and BYTE PTR [eax],ah

ca2: 20 20 and BYTE PTR [eax],ah

ca4: 69 66 20 28 73 70 20 imul esp,DWORD PTR [esi+0x20],0x20707328

cab: 3e 20 30 and BYTE PTR ds:[eax],dh

cae: 29 20 sub DWORD PTR [eax],esp

cb0: 7b 0a jnp 0xcbc

cb2: 20 20 and BYTE PTR [eax],ah

cb4: 20 20 and BYTE PTR [eax],ah

cb6: 20 20 and BYTE PTR [eax],ah

cb8: 20 20 and BYTE PTR [eax],ah

cba: 20 20 and BYTE PTR [eax],ah

cbc: 20 20 and BYTE PTR [eax],ah

cbe: 73 70 jae 0xd30

cc0: 2d 2d 3b 0a 20 sub eax,0x200a3b2d

cc5: 20 20 and BYTE PTR [eax],ah

cc7: 20 20 and BYTE PTR [eax],ah

cc9: 20 20 and BYTE PTR [eax],ah

ccb: 20 20 and BYTE PTR [eax],ah

ccd: 20 20 and BYTE PTR [eax],ah

ccf: 20 72 65 and BYTE PTR [edx+0x65],dh

cd2: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

cd9: 5b

cda: 72 65 jb 0xd41

cdc: 67 5d addr16 pop ebp

cde: 20 3d 20 6d 65 6d and BYTE PTR ds:0x6d656d20,bh

ce4: 6f outs dx,DWORD PTR ds:[esi]

ce5: 72 79 jb 0xd60

ce7: 5b pop ebx

ce8: 73 70 jae 0xd5a

cea: 5d pop ebp

ceb: 3b 20 cmp esp,DWORD PTR [eax]

ced: 2f das

cee: 2f das

cef: 20 52 65 and BYTE PTR [edx+0x65],dl

cf2: 74 72 je 0xd66

cf4: 69 65 76 65 20 76 61 imul esp,DWORD PTR [ebp+0x76],0x61762065

cfb: 6c ins BYTE PTR es:[edi],dx

cfc: 75 65 jne 0xd63

cfe: 20 66 72 and BYTE PTR [esi+0x72],ah

d01: 6f outs dx,DWORD PTR ds:[esi]

d02: 6d ins DWORD PTR es:[edi],dx

d03: 20 6d 65 and BYTE PTR [ebp+0x65],ch

d06: 6d ins DWORD PTR es:[edi],dx

d07: 6f outs dx,DWORD PTR ds:[esi]

d08: 72 79 jb 0xd83

d0a: 0a 20 or ah,BYTE PTR [eax]

d0c: 20 20 and BYTE PTR [eax],ah

d0e: 20 20 and BYTE PTR [eax],ah

d10: 20 20 and BYTE PTR [eax],ah

d12: 20 7d 0a and BYTE PTR [ebp+0xa],bh

d15: 20 20 and BYTE PTR [eax],ah

d17: 20 20 and BYTE PTR [eax],ah

d19: 7d 0a jge 0xd25

d1b: 0a 20 or ah,BYTE PTR [eax]

d1d: 20 20 and BYTE PTR [eax],ah

d1f: 20 2f and BYTE PTR [edi],ch

d21: 2f das

d22: 20 53 55 and BYTE PTR [ebx+0x55],dl

d25: 42 inc edx

d26: 20 69 6e and BYTE PTR [ecx+0x6e],ch

d29: 73 74 jae 0xd9f

d2b: 72 75 jb 0xda2

d2d: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

d31: 6e outs dx,BYTE PTR ds:[esi]

d32: 3a 20 cmp ah,BYTE PTR [eax]

d34: 53 push ebx

d35: 75 62 jne 0xd99

d37: 74 72 je 0xdab

d39: 61 popa

d3a: 63 74 20 72 arpl WORD PTR [eax+eiz\*1+0x72],si

d3e: 65 67 69 73 74 65 72 imul esi,DWORD PTR gs:[bp+di+0x74],0x76207265

d45: 20 76

d47: 61 popa

d48: 6c ins BYTE PTR es:[edi],dx

d49: 75 65 jne 0xdb0

d4b: 73 0a jae 0xd57

d4d: 20 20 and BYTE PTR [eax],ah

d4f: 20 20 and BYTE PTR [eax],ah

d51: 76 6f jbe 0xdc2

d53: 69 64 20 53 55 42 28 imul esp,DWORD PTR [eax+eiz\*1+0x53],0x69284255

d5a: 69

d5b: 6e outs dx,BYTE PTR ds:[esi]

d5c: 74 20 je 0xd7e

d5e: 72 65 jb 0xdc5

d60: 67 31 2c xor DWORD PTR [si],ebp

d63: 20 69 6e and BYTE PTR [ecx+0x6e],ch

d66: 74 20 je 0xd88

d68: 72 65 jb 0xdcf

d6a: 67 32 29 xor ch,BYTE PTR [bx+di]

d6d: 20 7b 0a and BYTE PTR [ebx+0xa],bh

d70: 20 20 and BYTE PTR [eax],ah

d72: 20 20 and BYTE PTR [eax],ah

d74: 20 20 and BYTE PTR [eax],ah

d76: 20 20 and BYTE PTR [eax],ah

d78: 72 65 jb 0xddf

d7a: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

d81: 5b

d82: 72 65 jb 0xde9

d84: 67 31 5d 20 xor DWORD PTR [di+0x20],ebx

d88: 3d 20 41 4c 55 cmp eax,0x554c4120

d8d: 28 72 65 sub BYTE PTR [edx+0x65],dh

d90: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

d97: 5b

d98: 72 65 jb 0xdff

d9a: 67 31 5d 2c xor DWORD PTR [di+0x2c],ebx

d9e: 20 72 65 and BYTE PTR [edx+0x65],dh

da1: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

da8: 5b

da9: 72 65 jb 0xe10

dab: 67 32 5d 2c xor bl,BYTE PTR [di+0x2c]

daf: 20 74 72 75 and BYTE PTR [edx+esi\*2+0x75],dh

db3: 65 2c 20 gs sub al,0x20

db6: 66 61 popaw

db8: 6c ins BYTE PTR es:[edi],dx

db9: 73 65 jae 0xe20

dbb: 2c 20 sub al,0x20

dbd: 66 61 popaw

dbf: 6c ins BYTE PTR es:[edi],dx

dc0: 73 65 jae 0xe27

dc2: 29 3b sub DWORD PTR [ebx],edi

dc4: 0a 20 or ah,BYTE PTR [eax]

dc6: 20 20 and BYTE PTR [eax],ah

dc8: 20 7d 0a and BYTE PTR [ebp+0xa],bh

dcb: 0a 20 or ah,BYTE PTR [eax]

dcd: 20 20 and BYTE PTR [eax],ah

dcf: 20 2f and BYTE PTR [edi],ch

dd1: 2f das

dd2: 20 41 44 and BYTE PTR [ecx+0x44],al

dd5: 44 inc esp

dd6: 20 69 6e and BYTE PTR [ecx+0x6e],ch

dd9: 73 74 jae 0xe4f

ddb: 72 75 jb 0xe52

ddd: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

de1: 6e outs dx,BYTE PTR ds:[esi]

de2: 3a 20 cmp ah,BYTE PTR [eax]

de4: 41 inc ecx

de5: 64 64 20 72 65 fs and BYTE PTR fs:[edx+0x65],dh

dea: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x76207265

df1: 76

df2: 61 popa

df3: 6c ins BYTE PTR es:[edi],dx

df4: 75 65 jne 0xe5b

df6: 73 0a jae 0xe02

df8: 20 20 and BYTE PTR [eax],ah

dfa: 20 20 and BYTE PTR [eax],ah

dfc: 76 6f jbe 0xe6d

dfe: 69 64 20 41 44 44 28 imul esp,DWORD PTR [eax+eiz\*1+0x41],0x69284444

e05: 69

e06: 6e outs dx,BYTE PTR ds:[esi]

e07: 74 20 je 0xe29

e09: 72 65 jb 0xe70

e0b: 67 31 2c xor DWORD PTR [si],ebp

e0e: 20 69 6e and BYTE PTR [ecx+0x6e],ch

e11: 74 20 je 0xe33

e13: 72 65 jb 0xe7a

e15: 67 32 29 xor ch,BYTE PTR [bx+di]

e18: 20 7b 0a and BYTE PTR [ebx+0xa],bh

e1b: 20 20 and BYTE PTR [eax],ah

e1d: 20 20 and BYTE PTR [eax],ah

e1f: 20 20 and BYTE PTR [eax],ah

e21: 20 20 and BYTE PTR [eax],ah

e23: 72 65 jb 0xe8a

e25: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

e2c: 5b

e2d: 72 65 jb 0xe94

e2f: 67 31 5d 20 xor DWORD PTR [di+0x20],ebx

e33: 3d 20 41 4c 55 cmp eax,0x554c4120

e38: 28 72 65 sub BYTE PTR [edx+0x65],dh

e3b: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

e42: 5b

e43: 72 65 jb 0xeaa

e45: 67 31 5d 2c xor DWORD PTR [di+0x2c],ebx

e49: 20 72 65 and BYTE PTR [edx+0x65],dh

e4c: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

e53: 5b

e54: 72 65 jb 0xebb

e56: 67 32 5d 2c xor bl,BYTE PTR [di+0x2c]

e5a: 20 66 61 and BYTE PTR [esi+0x61],ah

e5d: 6c ins BYTE PTR es:[edi],dx

e5e: 73 65 jae 0xec5

e60: 2c 20 sub al,0x20

e62: 66 61 popaw

e64: 6c ins BYTE PTR es:[edi],dx

e65: 73 65 jae 0xecc

e67: 2c 20 sub al,0x20

e69: 66 61 popaw

e6b: 6c ins BYTE PTR es:[edi],dx

e6c: 73 65 jae 0xed3

e6e: 29 3b sub DWORD PTR [ebx],edi

e70: 0a 20 or ah,BYTE PTR [eax]

e72: 20 20 and BYTE PTR [eax],ah

e74: 20 7d 0a and BYTE PTR [ebp+0xa],bh

e77: 0a 20 or ah,BYTE PTR [eax]

e79: 20 20 and BYTE PTR [eax],ah

e7b: 20 2f and BYTE PTR [edi],ch

e7d: 2f das

e7e: 20 4c 6f 63 and BYTE PTR [edi+ebp\*2+0x63],cl

e82: 6b 20 49 imul esp,DWORD PTR [eax],0x49

e85: 6e outs dx,BYTE PTR ds:[esi]

e86: 73 74 jae 0xefc

e88: 72 75 jb 0xeff

e8a: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

e8e: 6e outs dx,BYTE PTR ds:[esi]

e8f: 3a 20 cmp ah,BYTE PTR [eax]

e91: 46 inc esi

e92: 6f outs dx,DWORD PTR ds:[esi]

e93: 72 20 jb 0xeb5

e95: 6d ins DWORD PTR es:[edi],dx

e96: 75 6c jne 0xf04

e98: 74 69 je 0xf03

e9a: 2d 63 6f 72 65 sub eax,0x65726f63

e9f: 20 73 79 and BYTE PTR [ebx+0x79],dh

ea2: 6e outs dx,BYTE PTR ds:[esi]

ea3: 63 68 72 arpl WORD PTR [eax+0x72],bp

ea6: 6f outs dx,DWORD PTR ds:[esi]

ea7: 6e outs dx,BYTE PTR ds:[esi]

ea8: 69 7a 61 74 69 6f 6e imul edi,DWORD PTR [edx+0x61],0x6e6f6974

eaf: 0a 20 or ah,BYTE PTR [eax]

eb1: 20 20 and BYTE PTR [eax],ah

eb3: 20 76 6f and BYTE PTR [esi+0x6f],dh

eb6: 69 64 20 4c 4f 43 4b imul esp,DWORD PTR [eax+eiz\*1+0x4c],0x284b434f

ebd: 28

ebe: 29 20 sub DWORD PTR [eax],esp

ec0: 7b 0a jnp 0xecc

ec2: 20 20 and BYTE PTR [eax],ah

ec4: 20 20 and BYTE PTR [eax],ah

ec6: 20 20 and BYTE PTR [eax],ah

ec8: 20 20 and BYTE PTR [eax],ah

eca: 73 74 jae 0xf40

ecc: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

ecf: 6c ins BYTE PTR es:[edi],dx

ed0: 6f outs dx,DWORD PTR ds:[esi]

ed1: 63 6b 5f arpl WORD PTR [ebx+0x5f],bp

ed4: 67 75 61 addr16 jne 0xf38

ed7: 72 64 jb 0xf3d

ed9: 3c 73 cmp al,0x73

edb: 74 64 je 0xf41

edd: 3a 3a cmp bh,BYTE PTR [edx]

edf: 6d ins DWORD PTR es:[edi],dx

ee0: 75 74 jne 0xf56

ee2: 65 78 3e gs js 0xf23

ee5: 20 6c 6f 63 and BYTE PTR [edi+ebp\*2+0x63],ch

ee9: 6b 28 63 imul ebp,DWORD PTR [eax],0x63

eec: 6f outs dx,DWORD PTR ds:[esi]

eed: 72 65 jb 0xf54

eef: 5f pop edi

ef0: 6d ins DWORD PTR es:[edi],dx

ef1: 75 74 jne 0xf67

ef3: 65 78 29 gs js 0xf1f

ef6: 3b 0a cmp ecx,DWORD PTR [edx]

ef8: 20 20 and BYTE PTR [eax],ah

efa: 20 20 and BYTE PTR [eax],ah

efc: 7d 0a jge 0xf08

efe: 0a 20 or ah,BYTE PTR [eax]

f00: 20 20 and BYTE PTR [eax],ah

f02: 20 2f and BYTE PTR [edi],ch

f04: 2f das

f05: 20 58 43 and BYTE PTR [eax+0x43],bl

f08: 48 dec eax

f09: 47 inc edi

f0a: 20 49 6e and BYTE PTR [ecx+0x6e],cl

f0d: 73 74 jae 0xf83

f0f: 72 75 jb 0xf86

f11: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

f15: 6e outs dx,BYTE PTR ds:[esi]

f16: 3a 20 cmp ah,BYTE PTR [eax]

f18: 45 inc ebp

f19: 78 63 js 0xf7e

f1b: 68 61 6e 67 65 push 0x65676e61

f20: 20 72 65 and BYTE PTR [edx+0x65],dh

f23: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x76207265

f2a: 76

f2b: 61 popa

f2c: 6c ins BYTE PTR es:[edi],dx

f2d: 75 65 jne 0xf94

f2f: 73 20 jae 0xf51

f31: 28 66 6f sub BYTE PTR [esi+0x6f],ah

f34: 72 20 jb 0xf56

f36: 61 popa

f37: 74 6f je 0xfa8

f39: 6d ins DWORD PTR es:[edi],dx

f3a: 69 63 20 6f 70 65 72 imul esp,DWORD PTR [ebx+0x20],0x7265706f

f41: 61 popa

f42: 74 69 je 0xfad

f44: 6f outs dx,DWORD PTR ds:[esi]

f45: 6e outs dx,BYTE PTR ds:[esi]

f46: 73 29 jae 0xf71

f48: 0a 20 or ah,BYTE PTR [eax]

f4a: 20 20 and BYTE PTR [eax],ah

f4c: 20 76 6f and BYTE PTR [esi+0x6f],dh

f4f: 69 64 20 58 43 48 47 imul esp,DWORD PTR [eax+eiz\*1+0x58],0x28474843

f56: 28

f57: 69 6e 74 20 72 65 67 imul ebp,DWORD PTR [esi+0x74],0x67657220

f5e: 31 2c 20 xor DWORD PTR [eax+eiz\*1],ebp

f61: 69 6e 74 20 72 65 67 imul ebp,DWORD PTR [esi+0x74],0x67657220

f68: 32 29 xor ch,BYTE PTR [ecx]

f6a: 20 7b 0a and BYTE PTR [ebx+0xa],bh

f6d: 20 20 and BYTE PTR [eax],ah

f6f: 20 20 and BYTE PTR [eax],ah

f71: 20 20 and BYTE PTR [eax],ah

f73: 20 20 and BYTE PTR [eax],ah

f75: 73 74 jae 0xfeb

f77: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

f7a: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

f7d: 73 65 jae 0xfe4

f7f: 74 3c je 0xfbd

f81: 31 36 xor DWORD PTR [esi],esi

f83: 3e 20 74 65 6d and BYTE PTR ds:[ebp+eiz\*2+0x6d],dh

f88: 70 20 jo 0xfaa

f8a: 3d 20 72 65 67 cmp eax,0x67657220

f8f: 69 73 74 65 72 73 5b imul esi,DWORD PTR [ebx+0x74],0x5b737265

f96: 72 65 jb 0xffd

f98: 67 31 5d 3b xor DWORD PTR [di+0x3b],ebx

f9c: 0a 20 or ah,BYTE PTR [eax]

f9e: 20 20 and BYTE PTR [eax],ah

fa0: 20 20 and BYTE PTR [eax],ah

fa2: 20 20 and BYTE PTR [eax],ah

fa4: 20 72 65 and BYTE PTR [edx+0x65],dh

fa7: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

fae: 5b

faf: 72 65 jb 0x1016

fb1: 67 31 5d 20 xor DWORD PTR [di+0x20],ebx

fb5: 3d 20 72 65 67 cmp eax,0x67657220

fba: 69 73 74 65 72 73 5b imul esi,DWORD PTR [ebx+0x74],0x5b737265

fc1: 72 65 jb 0x1028

fc3: 67 32 5d 3b xor bl,BYTE PTR [di+0x3b]

fc7: 0a 20 or ah,BYTE PTR [eax]

fc9: 20 20 and BYTE PTR [eax],ah

fcb: 20 20 and BYTE PTR [eax],ah

fcd: 20 20 and BYTE PTR [eax],ah

fcf: 20 72 65 and BYTE PTR [edx+0x65],dh

fd2: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

fd9: 5b

fda: 72 65 jb 0x1041

fdc: 67 32 5d 20 xor bl,BYTE PTR [di+0x20]

fe0: 3d 20 74 65 6d cmp eax,0x6d657420

fe5: 70 3b jo 0x1022

fe7: 0a 20 or ah,BYTE PTR [eax]

fe9: 20 20 and BYTE PTR [eax],ah

feb: 20 7d 0a and BYTE PTR [ebp+0xa],bh

fee: 0a 20 or ah,BYTE PTR [eax]

ff0: 20 20 and BYTE PTR [eax],ah

ff2: 20 2f and BYTE PTR [edi],ch

ff4: 2f das

ff5: 20 43 4d and BYTE PTR [ebx+0x4d],al

ff8: 50 push eax

ff9: 58 pop eax

ffa: 43 inc ebx

ffb: 48 dec eax

ffc: 47 inc edi

ffd: 20 49 6e and BYTE PTR [ecx+0x6e],cl

1000: 73 74 jae 0x1076

1002: 72 75 jb 0x1079

1004: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

1008: 6e outs dx,BYTE PTR ds:[esi]

1009: 3a 20 cmp ah,BYTE PTR [eax]

100b: 43 inc ebx

100c: 6f outs dx,DWORD PTR ds:[esi]

100d: 6d ins DWORD PTR es:[edi],dx

100e: 70 61 jo 0x1071

1010: 72 65 jb 0x1077

1012: 20 61 6e and BYTE PTR [ecx+0x6e],ah

1015: 64 20 65 78 and BYTE PTR fs:[ebp+0x78],ah

1019: 63 68 61 arpl WORD PTR [eax+0x61],bp

101c: 6e outs dx,BYTE PTR ds:[esi]

101d: 67 65 20 72 65 and BYTE PTR gs:[bp+si+0x65],dh

1022: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x76207265

1029: 76

102a: 61 popa

102b: 6c ins BYTE PTR es:[edi],dx

102c: 75 65 jne 0x1093

102e: 73 0a jae 0x103a

1030: 20 20 and BYTE PTR [eax],ah

1032: 20 20 and BYTE PTR [eax],ah

1034: 76 6f jbe 0x10a5

1036: 69 64 20 43 4d 50 58 imul esp,DWORD PTR [eax+eiz\*1+0x43],0x4358504d

103d: 43

103e: 48 dec eax

103f: 47 inc edi

1040: 28 69 6e sub BYTE PTR [ecx+0x6e],ch

1043: 74 20 je 0x1065

1045: 72 65 jb 0x10ac

1047: 67 31 2c xor DWORD PTR [si],ebp

104a: 20 69 6e and BYTE PTR [ecx+0x6e],ch

104d: 74 20 je 0x106f

104f: 72 65 jb 0x10b6

1051: 67 32 29 xor ch,BYTE PTR [bx+di]

1054: 20 7b 0a and BYTE PTR [ebx+0xa],bh

1057: 20 20 and BYTE PTR [eax],ah

1059: 20 20 and BYTE PTR [eax],ah

105b: 20 20 and BYTE PTR [eax],ah

105d: 20 20 and BYTE PTR [eax],ah

105f: 69 66 20 28 72 65 67 imul esp,DWORD PTR [esi+0x20],0x67657228

1066: 69 73 74 65 72 73 5b imul esi,DWORD PTR [ebx+0x74],0x5b737265

106d: 72 65 jb 0x10d4

106f: 67 31 5d 20 xor DWORD PTR [di+0x20],ebx

1073: 3d 3d 20 72 65 cmp eax,0x6572203d

1078: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

107f: 5b

1080: 72 65 jb 0x10e7

1082: 67 32 5d 29 xor bl,BYTE PTR [di+0x29]

1086: 20 7b 0a and BYTE PTR [ebx+0xa],bh

1089: 20 20 and BYTE PTR [eax],ah

108b: 20 20 and BYTE PTR [eax],ah

108d: 20 20 and BYTE PTR [eax],ah

108f: 20 20 and BYTE PTR [eax],ah

1091: 20 20 and BYTE PTR [eax],ah

1093: 20 20 and BYTE PTR [eax],ah

1095: 72 65 jb 0x10fc

1097: 67 69 73 74 65 72 73 imul esi,DWORD PTR [bp+di+0x74],0x5b737265

109e: 5b

109f: 72 65 jb 0x1106

10a1: 67 31 5d 20 xor DWORD PTR [di+0x20],ebx

10a5: 3d 20 72 65 67 cmp eax,0x67657220

10aa: 69 73 74 65 72 73 5b imul esi,DWORD PTR [ebx+0x74],0x5b737265

10b1: 72 65 jb 0x1118

10b3: 67 32 5d 3b xor bl,BYTE PTR [di+0x3b]

10b7: 20 2f and BYTE PTR [edi],ch

10b9: 2f das

10ba: 20 45 78 and BYTE PTR [ebp+0x78],al

10bd: 63 68 61 arpl WORD PTR [eax+0x61],bp

10c0: 6e outs dx,BYTE PTR ds:[esi]

10c1: 67 65 20 69 66 and BYTE PTR gs:[bx+di+0x66],ch

10c6: 20 65 71 and BYTE PTR [ebp+0x71],ah

10c9: 75 61 jne 0x112c

10cb: 6c ins BYTE PTR es:[edi],dx

10cc: 0a 20 or ah,BYTE PTR [eax]

10ce: 20 20 and BYTE PTR [eax],ah

10d0: 20 20 and BYTE PTR [eax],ah

10d2: 20 20 and BYTE PTR [eax],ah

10d4: 20 7d 0a and BYTE PTR [ebp+0xa],bh

10d7: 20 20 and BYTE PTR [eax],ah

10d9: 20 20 and BYTE PTR [eax],ah

10db: 7d 0a jge 0x10e7

10dd: 0a 20 or ah,BYTE PTR [eax]

10df: 20 20 and BYTE PTR [eax],ah

10e1: 20 2f and BYTE PTR [edi],ch

10e3: 2f das

10e4: 20 43 4c and BYTE PTR [ebx+0x4c],al

10e7: 49 dec ecx

10e8: 20 49 6e and BYTE PTR [ecx+0x6e],cl

10eb: 73 74 jae 0x1161

10ed: 72 75 jb 0x1164

10ef: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

10f3: 6e outs dx,BYTE PTR ds:[esi]

10f4: 3a 20 cmp ah,BYTE PTR [eax]

10f6: 43 inc ebx

10f7: 6c ins BYTE PTR es:[edi],dx

10f8: 65 61 gs popa

10fa: 72 20 jb 0x111c

10fc: 69 6e 74 65 72 72 75 imul ebp,DWORD PTR [esi+0x74],0x75727265

1103: 70 74 jo 0x1179

1105: 20 66 6c and BYTE PTR [esi+0x6c],ah

1108: 61 popa

1109: 67 0a 20 or ah,BYTE PTR [bx+si]

110c: 20 20 and BYTE PTR [eax],ah

110e: 20 76 6f and BYTE PTR [esi+0x6f],dh

1111: 69 64 20 43 4c 49 28 imul esp,DWORD PTR [eax+eiz\*1+0x43],0x2928494c

1118: 29

1119: 20 7b 0a and BYTE PTR [ebx+0xa],bh

111c: 20 20 and BYTE PTR [eax],ah

111e: 20 20 and BYTE PTR [eax],ah

1120: 20 20 and BYTE PTR [eax],ah

1122: 20 20 and BYTE PTR [eax],ah

1124: 2f das

1125: 2f das

1126: 20 49 6e and BYTE PTR [ecx+0x6e],cl

1129: 20 61 20 and BYTE PTR [ecx+0x20],ah

112c: 72 65 jb 0x1193

112e: 61 popa

112f: 6c ins BYTE PTR es:[edi],dx

1130: 20 43 50 and BYTE PTR [ebx+0x50],al

1133: 55 push ebp

1134: 2c 20 sub al,0x20

1136: 74 68 je 0x11a0

1138: 69 73 20 77 6f 75 6c imul esi,DWORD PTR [ebx+0x20],0x6c756f77

113f: 64 20 64 69 73 and BYTE PTR fs:[ecx+ebp\*2+0x73],ah

1144: 61 popa

1145: 62 6c 65 20 bound ebp,QWORD PTR [ebp+eiz\*2+0x20]

1149: 69 6e 74 65 72 72 75 imul ebp,DWORD PTR [esi+0x74],0x75727265

1150: 70 74 jo 0x11c6

1152: 73 2e jae 0x1182

1154: 0a 20 or ah,BYTE PTR [eax]

1156: 20 20 and BYTE PTR [eax],ah

1158: 20 20 and BYTE PTR [eax],ah

115a: 20 20 and BYTE PTR [eax],ah

115c: 20 73 74 and BYTE PTR [ebx+0x74],dh

115f: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1162: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1165: 74 20 je 0x1187

1167: 3c 3c cmp al,0x3c

1169: 20 22 and BYTE PTR [edx],ah

116b: 49 dec ecx

116c: 6e outs dx,BYTE PTR ds:[esi]

116d: 74 65 je 0x11d4

116f: 72 72 jb 0x11e3

1171: 75 70 jne 0x11e3

1173: 74 20 je 0x1195

1175: 66 6c data16 ins BYTE PTR es:[edi],dx

1177: 61 popa

1178: 67 20 63 6c and BYTE PTR [bp+di+0x6c],ah

117c: 65 61 gs popa

117e: 72 65 jb 0x11e5

1180: 64 2e 5c fs cs pop esp

1183: 6e outs dx,BYTE PTR ds:[esi]

1184: 22 3b and bh,BYTE PTR [ebx]

1186: 0a 20 or ah,BYTE PTR [eax]

1188: 20 20 and BYTE PTR [eax],ah

118a: 20 7d 0a and BYTE PTR [ebp+0xa],bh

118d: 0a 20 or ah,BYTE PTR [eax]

118f: 20 20 and BYTE PTR [eax],ah

1191: 20 2f and BYTE PTR [edi],ch

1193: 2f das

1194: 20 53 54 and BYTE PTR [ebx+0x54],dl

1197: 49 dec ecx

1198: 20 49 6e and BYTE PTR [ecx+0x6e],cl

119b: 73 74 jae 0x1211

119d: 72 75 jb 0x1214

119f: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

11a3: 6e outs dx,BYTE PTR ds:[esi]

11a4: 3a 20 cmp ah,BYTE PTR [eax]

11a6: 53 push ebx

11a7: 65 74 20 gs je 0x11ca

11aa: 69 6e 74 65 72 72 75 imul ebp,DWORD PTR [esi+0x74],0x75727265

11b1: 70 74 jo 0x1227

11b3: 20 66 6c and BYTE PTR [esi+0x6c],ah

11b6: 61 popa

11b7: 67 0a 20 or ah,BYTE PTR [bx+si]

11ba: 20 20 and BYTE PTR [eax],ah

11bc: 20 76 6f and BYTE PTR [esi+0x6f],dh

11bf: 69 64 20 53 54 49 28 imul esp,DWORD PTR [eax+eiz\*1+0x53],0x29284954

11c6: 29

11c7: 20 7b 0a and BYTE PTR [ebx+0xa],bh

11ca: 20 20 and BYTE PTR [eax],ah

11cc: 20 20 and BYTE PTR [eax],ah

11ce: 20 20 and BYTE PTR [eax],ah

11d0: 20 20 and BYTE PTR [eax],ah

11d2: 2f das

11d3: 2f das

11d4: 20 49 6e and BYTE PTR [ecx+0x6e],cl

11d7: 20 61 20 and BYTE PTR [ecx+0x20],ah

11da: 72 65 jb 0x1241

11dc: 61 popa

11dd: 6c ins BYTE PTR es:[edi],dx

11de: 20 43 50 and BYTE PTR [ebx+0x50],al

11e1: 55 push ebp

11e2: 2c 20 sub al,0x20

11e4: 74 68 je 0x124e

11e6: 69 73 20 77 6f 75 6c imul esi,DWORD PTR [ebx+0x20],0x6c756f77

11ed: 64 20 65 6e and BYTE PTR fs:[ebp+0x6e],ah

11f1: 61 popa

11f2: 62 6c 65 20 bound ebp,QWORD PTR [ebp+eiz\*2+0x20]

11f6: 69 6e 74 65 72 72 75 imul ebp,DWORD PTR [esi+0x74],0x75727265

11fd: 70 74 jo 0x1273

11ff: 73 2e jae 0x122f

1201: 0a 20 or ah,BYTE PTR [eax]

1203: 20 20 and BYTE PTR [eax],ah

1205: 20 20 and BYTE PTR [eax],ah

1207: 20 20 and BYTE PTR [eax],ah

1209: 20 73 74 and BYTE PTR [ebx+0x74],dh

120c: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

120f: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1212: 74 20 je 0x1234

1214: 3c 3c cmp al,0x3c

1216: 20 22 and BYTE PTR [edx],ah

1218: 49 dec ecx

1219: 6e outs dx,BYTE PTR ds:[esi]

121a: 74 65 je 0x1281

121c: 72 72 jb 0x1290

121e: 75 70 jne 0x1290

1220: 74 20 je 0x1242

1222: 66 6c data16 ins BYTE PTR es:[edi],dx

1224: 61 popa

1225: 67 20 73 65 and BYTE PTR [bp+di+0x65],dh

1229: 74 2e je 0x1259

122b: 5c pop esp

122c: 6e outs dx,BYTE PTR ds:[esi]

122d: 22 3b and bh,BYTE PTR [ebx]

122f: 0a 20 or ah,BYTE PTR [eax]

1231: 20 20 and BYTE PTR [eax],ah

1233: 20 7d 0a and BYTE PTR [ebp+0xa],bh

1236: 0a 20 or ah,BYTE PTR [eax]

1238: 20 20 and BYTE PTR [eax],ah

123a: 20 2f and BYTE PTR [edi],ch

123c: 2f das

123d: 20 53 46 and BYTE PTR [ebx+0x46],dl

1240: 45 inc ebp

1241: 4e dec esi

1242: 43 inc ebx

1243: 45 inc ebp

1244: 20 49 6e and BYTE PTR [ecx+0x6e],cl

1247: 73 74 jae 0x12bd

1249: 72 75 jb 0x12c0

124b: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

124f: 6e outs dx,BYTE PTR ds:[esi]

1250: 3a 20 cmp ah,BYTE PTR [eax]

1252: 4d dec ebp

1253: 65 6d gs ins DWORD PTR es:[edi],dx

1255: 6f outs dx,DWORD PTR ds:[esi]

1256: 72 79 jb 0x12d1

1258: 20 66 65 and BYTE PTR [esi+0x65],ah

125b: 6e outs dx,BYTE PTR ds:[esi]

125c: 63 65 20 arpl WORD PTR [ebp+0x20],sp

125f: 74 6f je 0x12d0

1261: 20 65 6e and BYTE PTR [ebp+0x6e],ah

1264: 73 75 jae 0x12db

1266: 72 65 jb 0x12cd

1268: 20 73 74 and BYTE PTR [ebx+0x74],dh

126b: 6f outs dx,DWORD PTR ds:[esi]

126c: 72 65 jb 0x12d3

126e: 73 20 jae 0x1290

1270: 61 popa

1271: 72 65 jb 0x12d8

1273: 20 6f 72 and BYTE PTR [edi+0x72],ch

1276: 64 65 72 65 fs gs jb 0x12df

127a: 64 0a 20 or ah,BYTE PTR fs:[eax]

127d: 20 20 and BYTE PTR [eax],ah

127f: 20 76 6f and BYTE PTR [esi+0x6f],dh

1282: 69 64 20 53 46 45 4e imul esp,DWORD PTR [eax+eiz\*1+0x53],0x434e4546

1289: 43

128a: 45 inc ebp

128b: 28 29 sub BYTE PTR [ecx],ch

128d: 20 7b 0a and BYTE PTR [ebx+0xa],bh

1290: 20 20 and BYTE PTR [eax],ah

1292: 20 20 and BYTE PTR [eax],ah

1294: 20 20 and BYTE PTR [eax],ah

1296: 20 20 and BYTE PTR [eax],ah

1298: 2f das

1299: 2f das

129a: 20 49 6e and BYTE PTR [ecx+0x6e],cl

129d: 20 61 20 and BYTE PTR [ecx+0x20],ah

12a0: 72 65 jb 0x1307

12a2: 61 popa

12a3: 6c ins BYTE PTR es:[edi],dx

12a4: 20 43 50 and BYTE PTR [ebx+0x50],al

12a7: 55 push ebp

12a8: 2c 20 sub al,0x20

12aa: 74 68 je 0x1314

12ac: 69 73 20 77 6f 75 6c imul esi,DWORD PTR [ebx+0x20],0x6c756f77

12b3: 64 20 65 6e and BYTE PTR fs:[ebp+0x6e],ah

12b7: 66 6f outs dx,WORD PTR ds:[esi]

12b9: 72 63 jb 0x131e

12bb: 65 20 6d 65 and BYTE PTR gs:[ebp+0x65],ch

12bf: 6d ins DWORD PTR es:[edi],dx

12c0: 6f outs dx,DWORD PTR ds:[esi]

12c1: 72 79 jb 0x133c

12c3: 20 6f 72 and BYTE PTR [edi+0x72],ch

12c6: 64 65 72 69 fs gs jb 0x1333

12ca: 6e outs dx,BYTE PTR ds:[esi]

12cb: 67 20 62 65 and BYTE PTR [bp+si+0x65],ah

12cf: 74 77 je 0x1348

12d1: 65 65 6e gs outs dx,BYTE PTR gs:[esi]

12d4: 20 63 6f and BYTE PTR [ebx+0x6f],ah

12d7: 72 65 jb 0x133e

12d9: 73 2e jae 0x1309

12db: 0a 20 or ah,BYTE PTR [eax]

12dd: 20 20 and BYTE PTR [eax],ah

12df: 20 20 and BYTE PTR [eax],ah

12e1: 20 20 and BYTE PTR [eax],ah

12e3: 20 73 74 and BYTE PTR [ebx+0x74],dh

12e6: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

12e9: 63 6f 75 arpl WORD PTR [edi+0x75],bp

12ec: 74 20 je 0x130e

12ee: 3c 3c cmp al,0x3c

12f0: 20 22 and BYTE PTR [edx],ah

12f2: 53 push ebx

12f3: 74 6f je 0x1364

12f5: 72 65 jb 0x135c

12f7: 20 66 65 and BYTE PTR [esi+0x65],ah

12fa: 6e outs dx,BYTE PTR ds:[esi]

12fb: 63 65 20 arpl WORD PTR [ebp+0x20],sp

12fe: 61 popa

12ff: 70 70 jo 0x1371

1301: 6c ins BYTE PTR es:[edi],dx

1302: 69 65 64 2e 5c 6e 22 imul esp,DWORD PTR [ebp+0x64],0x226e5c2e

1309: 3b 0a cmp ecx,DWORD PTR [edx]

130b: 20 20 and BYTE PTR [eax],ah

130d: 20 20 and BYTE PTR [eax],ah

130f: 7d 0a jge 0x131b

1311: 0a 20 or ah,BYTE PTR [eax]

1313: 20 20 and BYTE PTR [eax],ah

1315: 20 2f and BYTE PTR [edi],ch

1317: 2f das

1318: 20 4c 46 45 and BYTE PTR [esi+eax\*2+0x45],cl

131c: 4e dec esi

131d: 43 inc ebx

131e: 45 inc ebp

131f: 20 49 6e and BYTE PTR [ecx+0x6e],cl

1322: 73 74 jae 0x1398

1324: 72 75 jb 0x139b

1326: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

132a: 6e outs dx,BYTE PTR ds:[esi]

132b: 3a 20 cmp ah,BYTE PTR [eax]

132d: 4d dec ebp

132e: 65 6d gs ins DWORD PTR es:[edi],dx

1330: 6f outs dx,DWORD PTR ds:[esi]

1331: 72 79 jb 0x13ac

1333: 20 66 65 and BYTE PTR [esi+0x65],ah

1336: 6e outs dx,BYTE PTR ds:[esi]

1337: 63 65 20 arpl WORD PTR [ebp+0x20],sp

133a: 74 6f je 0x13ab

133c: 20 65 6e and BYTE PTR [ebp+0x6e],ah

133f: 73 75 jae 0x13b6

1341: 72 65 jb 0x13a8

1343: 20 6c 6f 61 and BYTE PTR [edi+ebp\*2+0x61],ch

1347: 64 73 20 fs jae 0x136a

134a: 61 popa

134b: 72 65 jb 0x13b2

134d: 20 6f 72 and BYTE PTR [edi+0x72],ch

1350: 64 65 72 65 fs gs jb 0x13b9

1354: 64 0a 20 or ah,BYTE PTR fs:[eax]

1357: 20 20 and BYTE PTR [eax],ah

1359: 20 76 6f and BYTE PTR [esi+0x6f],dh

135c: 69 64 20 4c 46 45 4e imul esp,DWORD PTR [eax+eiz\*1+0x4c],0x434e4546

1363: 43

1364: 45 inc ebp

1365: 28 29 sub BYTE PTR [ecx],ch

1367: 20 7b 0a and BYTE PTR [ebx+0xa],bh

136a: 20 20 and BYTE PTR [eax],ah

136c: 20 20 and BYTE PTR [eax],ah

136e: 20 20 and BYTE PTR [eax],ah

1370: 20 20 and BYTE PTR [eax],ah

1372: 2f das

1373: 2f das

1374: 20 49 6e and BYTE PTR [ecx+0x6e],cl

1377: 20 61 20 and BYTE PTR [ecx+0x20],ah

137a: 72 65 jb 0x13e1

137c: 61 popa

137d: 6c ins BYTE PTR es:[edi],dx

137e: 20 43 50 and BYTE PTR [ebx+0x50],al

1381: 55 push ebp

1382: 2c 20 sub al,0x20

1384: 74 68 je 0x13ee

1386: 69 73 20 77 6f 75 6c imul esi,DWORD PTR [ebx+0x20],0x6c756f77

138d: 64 20 65 6e and BYTE PTR fs:[ebp+0x6e],ah

1391: 66 6f outs dx,WORD PTR ds:[esi]

1393: 72 63 jb 0x13f8

1395: 65 20 6c 6f 61 and BYTE PTR gs:[edi+ebp\*2+0x61],ch

139a: 64 20 6f 72 and BYTE PTR fs:[edi+0x72],ch

139e: 64 65 72 69 fs gs jb 0x140b

13a2: 6e outs dx,BYTE PTR ds:[esi]

13a3: 67 20 62 65 and BYTE PTR [bp+si+0x65],ah

13a7: 74 77 je 0x1420

13a9: 65 65 6e gs outs dx,BYTE PTR gs:[esi]

13ac: 20 63 6f and BYTE PTR [ebx+0x6f],ah

13af: 72 65 jb 0x1416

13b1: 73 2e jae 0x13e1

13b3: 0a 20 or ah,BYTE PTR [eax]

13b5: 20 20 and BYTE PTR [eax],ah

13b7: 20 20 and BYTE PTR [eax],ah

13b9: 20 20 and BYTE PTR [eax],ah

13bb: 20 73 74 and BYTE PTR [ebx+0x74],dh

13be: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

13c1: 63 6f 75 arpl WORD PTR [edi+0x75],bp

13c4: 74 20 je 0x13e6

13c6: 3c 3c cmp al,0x3c

13c8: 20 22 and BYTE PTR [edx],ah

13ca: 4c dec esp

13cb: 6f outs dx,DWORD PTR ds:[esi]

13cc: 61 popa

13cd: 64 20 66 65 and BYTE PTR fs:[esi+0x65],ah

13d1: 6e outs dx,BYTE PTR ds:[esi]

13d2: 63 65 20 arpl WORD PTR [ebp+0x20],sp

13d5: 61 popa

13d6: 70 70 jo 0x1448

13d8: 6c ins BYTE PTR es:[edi],dx

13d9: 69 65 64 2e 5c 6e 22 imul esp,DWORD PTR [ebp+0x64],0x226e5c2e

13e0: 3b 0a cmp ecx,DWORD PTR [edx]

13e2: 20 20 and BYTE PTR [eax],ah

13e4: 20 20 and BYTE PTR [eax],ah

13e6: 7d 0a jge 0x13f2

13e8: 0a 20 or ah,BYTE PTR [eax]

13ea: 20 20 and BYTE PTR [eax],ah

13ec: 20 2f and BYTE PTR [edi],ch

13ee: 2f das

13ef: 20 4d 46 and BYTE PTR [ebp+0x46],cl

13f2: 45 inc ebp

13f3: 4e dec esi

13f4: 43 inc ebx

13f5: 45 inc ebp

13f6: 20 49 6e and BYTE PTR [ecx+0x6e],cl

13f9: 73 74 jae 0x146f

13fb: 72 75 jb 0x1472

13fd: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

1401: 6e outs dx,BYTE PTR ds:[esi]

1402: 3a 20 cmp ah,BYTE PTR [eax]

1404: 4d dec ebp

1405: 65 6d gs ins DWORD PTR es:[edi],dx

1407: 6f outs dx,DWORD PTR ds:[esi]

1408: 72 79 jb 0x1483

140a: 20 66 65 and BYTE PTR [esi+0x65],ah

140d: 6e outs dx,BYTE PTR ds:[esi]

140e: 63 65 20 arpl WORD PTR [ebp+0x20],sp

1411: 74 6f je 0x1482

1413: 20 65 6e and BYTE PTR [ebp+0x6e],ah

1416: 73 75 jae 0x148d

1418: 72 65 jb 0x147f

141a: 20 61 6c and BYTE PTR [ecx+0x6c],ah

141d: 6c ins BYTE PTR es:[edi],dx

141e: 20 6d 65 and BYTE PTR [ebp+0x65],ch

1421: 6d ins DWORD PTR es:[edi],dx

1422: 6f outs dx,DWORD PTR ds:[esi]

1423: 72 79 jb 0x149e

1425: 20 6f 70 and BYTE PTR [edi+0x70],ch

1428: 65 72 61 gs jb 0x148c

142b: 74 69 je 0x1496

142d: 6f outs dx,DWORD PTR ds:[esi]

142e: 6e outs dx,BYTE PTR ds:[esi]

142f: 73 20 jae 0x1451

1431: 61 popa

1432: 72 65 jb 0x1499

1434: 20 6f 72 and BYTE PTR [edi+0x72],ch

1437: 64 65 72 65 fs gs jb 0x14a0

143b: 64 0a 20 or ah,BYTE PTR fs:[eax]

143e: 20 20 and BYTE PTR [eax],ah

1440: 20 76 6f and BYTE PTR [esi+0x6f],dh

1443: 69 64 20 4d 46 45 4e imul esp,DWORD PTR [eax+eiz\*1+0x4d],0x434e4546

144a: 43

144b: 45 inc ebp

144c: 28 29 sub BYTE PTR [ecx],ch

144e: 20 7b 0a and BYTE PTR [ebx+0xa],bh

1451: 20 20 and BYTE PTR [eax],ah

1453: 20 20 and BYTE PTR [eax],ah

1455: 20 20 and BYTE PTR [eax],ah

1457: 20 20 and BYTE PTR [eax],ah

1459: 2f das

145a: 2f das

145b: 20 49 6e and BYTE PTR [ecx+0x6e],cl

145e: 20 61 20 and BYTE PTR [ecx+0x20],ah

1461: 72 65 jb 0x14c8

1463: 61 popa

1464: 6c ins BYTE PTR es:[edi],dx

1465: 20 43 50 and BYTE PTR [ebx+0x50],al

1468: 55 push ebp

1469: 2c 20 sub al,0x20

146b: 74 68 je 0x14d5

146d: 69 73 20 77 6f 75 6c imul esi,DWORD PTR [ebx+0x20],0x6c756f77

1474: 64 20 65 6e and BYTE PTR fs:[ebp+0x6e],ah

1478: 73 75 jae 0x14ef

147a: 72 65 jb 0x14e1

147c: 20 74 68 61 and BYTE PTR [eax+ebp\*2+0x61],dh

1480: 74 20 je 0x14a2

1482: 61 popa

1483: 6c ins BYTE PTR es:[edi],dx

1484: 6c ins BYTE PTR es:[edi],dx

1485: 20 6d 65 and BYTE PTR [ebp+0x65],ch

1488: 6d ins DWORD PTR es:[edi],dx

1489: 6f outs dx,DWORD PTR ds:[esi]

148a: 72 79 jb 0x1505

148c: 20 6f 70 and BYTE PTR [edi+0x70],ch

148f: 65 72 61 gs jb 0x14f3

1492: 74 69 je 0x14fd

1494: 6f outs dx,DWORD PTR ds:[esi]

1495: 6e outs dx,BYTE PTR ds:[esi]

1496: 73 20 jae 0x14b8

1498: 61 popa

1499: 72 65 jb 0x1500

149b: 20 63 6f and BYTE PTR [ebx+0x6f],ah

149e: 6d ins DWORD PTR es:[edi],dx

149f: 70 6c jo 0x150d

14a1: 65 74 65 gs je 0x1509

14a4: 20 62 65 and BYTE PTR [edx+0x65],ah

14a7: 66 6f outs dx,WORD PTR ds:[esi]

14a9: 72 65 jb 0x1510

14ab: 20 70 72 and BYTE PTR [eax+0x72],dh

14ae: 6f outs dx,DWORD PTR ds:[esi]

14af: 63 65 65 arpl WORD PTR [ebp+0x65],sp

14b2: 64 69 6e 67 2e 0a 20 imul ebp,DWORD PTR fs:[esi+0x67],0x20200a2e

14b9: 20

14ba: 20 20 and BYTE PTR [eax],ah

14bc: 20 20 and BYTE PTR [eax],ah

14be: 20 20 and BYTE PTR [eax],ah

14c0: 73 74 jae 0x1536

14c2: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

14c5: 63 6f 75 arpl WORD PTR [edi+0x75],bp

14c8: 74 20 je 0x14ea

14ca: 3c 3c cmp al,0x3c

14cc: 20 22 and BYTE PTR [edx],ah

14ce: 4d dec ebp

14cf: 65 6d gs ins DWORD PTR es:[edi],dx

14d1: 6f outs dx,DWORD PTR ds:[esi]

14d2: 72 79 jb 0x154d

14d4: 20 66 65 and BYTE PTR [esi+0x65],ah

14d7: 6e outs dx,BYTE PTR ds:[esi]

14d8: 63 65 20 arpl WORD PTR [ebp+0x20],sp

14db: 61 popa

14dc: 70 70 jo 0x154e

14de: 6c ins BYTE PTR es:[edi],dx

14df: 69 65 64 2e 5c 6e 22 imul esp,DWORD PTR [ebp+0x64],0x226e5c2e

14e6: 3b 0a cmp ecx,DWORD PTR [edx]

14e8: 20 20 and BYTE PTR [eax],ah

14ea: 20 20 and BYTE PTR [eax],ah

14ec: 7d 0a jge 0x14f8

14ee: 7d 3b jge 0x152b

14f0: 0a 0a or cl,BYTE PTR [edx]

14f2: 2f das

14f3: 2f das

14f4: 20 53 69 and BYTE PTR [ebx+0x69],dl

14f7: 6d ins DWORD PTR es:[edi],dx

14f8: 75 6c jne 0x1566

14fa: 61 popa

14fb: 74 65 je 0x1562

14fd: 20 45 78 and BYTE PTR [ebp+0x78],al

1500: 65 63 75 74 arpl WORD PTR gs:[ebp+0x74],si

1504: 69 6f 6e 20 66 6f 72 imul ebp,DWORD PTR [edi+0x6e],0x726f6620

150b: 20 61 20 and BYTE PTR [ecx+0x20],ah

150e: 43 inc ebx

150f: 6f outs dx,DWORD PTR ds:[esi]

1510: 72 65 jb 0x1577

1512: 0a 76 6f or dh,BYTE PTR [esi+0x6f]

1515: 69 64 20 65 78 65 63 imul esp,DWORD PTR [eax+eiz\*1+0x65],0x75636578

151c: 75

151d: 74 65 je 0x1584

151f: 5f pop edi

1520: 63 6f 72 arpl WORD PTR [edi+0x72],bp

1523: 65 28 43 50 sub BYTE PTR gs:[ebx+0x50],al

1527: 55 push ebp

1528: 20 26 and BYTE PTR [esi],ah

152a: 63 70 75 arpl WORD PTR [eax+0x75],si

152d: 2c 20 sub al,0x20

152f: 69 6e 74 20 63 6f 72 imul ebp,DWORD PTR [esi+0x74],0x726f6320

1536: 65 5f gs pop edi

1538: 69 64 29 20 7b 0a 20 imul esp,DWORD PTR [ecx+ebp\*1+0x20],0x20200a7b

153f: 20

1540: 20 20 and BYTE PTR [eax],ah

1542: 73 74 jae 0x15b8

1544: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1547: 6c ins BYTE PTR es:[edi],dx

1548: 6f outs dx,DWORD PTR ds:[esi]

1549: 63 6b 5f arpl WORD PTR [ebx+0x5f],bp

154c: 67 75 61 addr16 jne 0x15b0

154f: 72 64 jb 0x15b5

1551: 3c 73 cmp al,0x73

1553: 74 64 je 0x15b9

1555: 3a 3a cmp bh,BYTE PTR [edx]

1557: 6d ins DWORD PTR es:[edi],dx

1558: 75 74 jne 0x15ce

155a: 65 78 3e gs js 0x159b

155d: 20 6c 6f 63 and BYTE PTR [edi+ebp\*2+0x63],ch

1561: 6b 28 63 imul ebp,DWORD PTR [eax],0x63

1564: 6f outs dx,DWORD PTR ds:[esi]

1565: 72 65 jb 0x15cc

1567: 5f pop edi

1568: 6d ins DWORD PTR es:[edi],dx

1569: 75 74 jne 0x15df

156b: 65 78 29 gs js 0x1597

156e: 3b 0a cmp ecx,DWORD PTR [edx]

1570: 20 20 and BYTE PTR [eax],ah

1572: 20 20 and BYTE PTR [eax],ah

1574: 73 74 jae 0x15ea

1576: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1579: 63 6f 75 arpl WORD PTR [edi+0x75],bp

157c: 74 20 je 0x159e

157e: 3c 3c cmp al,0x3c

1580: 20 22 and BYTE PTR [edx],ah

1582: 45 inc ebp

1583: 78 65 js 0x15ea

1585: 63 75 74 arpl WORD PTR [ebp+0x74],si

1588: 69 6e 67 20 6f 6e 20 imul ebp,DWORD PTR [esi+0x67],0x206e6f20

158f: 63 6f 72 arpl WORD PTR [edi+0x72],bp

1592: 65 20 22 and BYTE PTR gs:[edx],ah

1595: 20 3c 3c and BYTE PTR [esp+edi\*1],bh

1598: 20 63 6f and BYTE PTR [ebx+0x6f],ah

159b: 72 65 jb 0x1602

159d: 5f pop edi

159e: 69 64 20 3c 3c 20 22 imul esp,DWORD PTR [eax+eiz\*1+0x3c],0x5c22203c

15a5: 5c

15a6: 6e outs dx,BYTE PTR ds:[esi]

15a7: 22 3b and bh,BYTE PTR [ebx]

15a9: 0a 0a or cl,BYTE PTR [edx]

15ab: 20 20 and BYTE PTR [eax],ah

15ad: 20 20 and BYTE PTR [eax],ah

15af: 2f das

15b0: 2f das

15b1: 20 4d 4f and BYTE PTR [ebp+0x4f],cl

15b4: 56 push esi

15b5: 20 69 6e and BYTE PTR [ecx+0x6e],ch

15b8: 73 74 jae 0x162e

15ba: 72 75 jb 0x1631

15bc: 63 74 69 6f arpl WORD PTR [ecx+ebp\*2+0x6f],si

15c0: 6e outs dx,BYTE PTR ds:[esi]

15c1: 20 65 78 and BYTE PTR [ebp+0x78],ah

15c4: 61 popa

15c5: 6d ins DWORD PTR es:[edi],dx

15c6: 70 6c jo 0x1634

15c8: 65 0a 20 or ah,BYTE PTR gs:[eax]

15cb: 20 20 and BYTE PTR [eax],ah

15cd: 20 63 70 and BYTE PTR [ebx+0x70],ah

15d0: 75 2e jne 0x1600

15d2: 4d dec ebp

15d3: 4f dec edi

15d4: 56 push esi

15d5: 28 30 sub BYTE PTR [eax],dh

15d7: 2c 20 sub al,0x20

15d9: 73 74 jae 0x164f

15db: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

15de: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

15e1: 73 65 jae 0x1648

15e3: 74 3c je 0x1621

15e5: 31 36 xor DWORD PTR [esi],esi

15e7: 3e 28 22 sub BYTE PTR ds:[edx],ah

15ea: 30 30 xor BYTE PTR [eax],dh

15ec: 30 30 xor BYTE PTR [eax],dh

15ee: 30 30 xor BYTE PTR [eax],dh

15f0: 30 30 xor BYTE PTR [eax],dh

15f2: 30 30 xor BYTE PTR [eax],dh

15f4: 30 30 xor BYTE PTR [eax],dh

15f6: 31 30 xor DWORD PTR [eax],esi

15f8: 31 30 xor DWORD PTR [eax],esi

15fa: 22 29 and ch,BYTE PTR [ecx]

15fc: 29 3b sub DWORD PTR [ebx],edi

15fe: 20 2f and BYTE PTR [edi],ch

1600: 2f das

1601: 20 4d 4f and BYTE PTR [ebp+0x4f],cl

1604: 56 push esi

1605: 20 31 and BYTE PTR [ecx],dh

1607: 30 20 xor BYTE PTR [eax],ah

1609: 74 6f je 0x167a

160b: 20 72 65 and BYTE PTR [edx+0x65],dh

160e: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x30207265

1615: 30

1616: 0a 20 or ah,BYTE PTR [eax]

1618: 20 20 and BYTE PTR [eax],ah

161a: 20 63 70 and BYTE PTR [ebx+0x70],ah

161d: 75 2e jne 0x164d

161f: 4d dec ebp

1620: 4f dec edi

1621: 56 push esi

1622: 28 31 sub BYTE PTR [ecx],dh

1624: 2c 20 sub al,0x20

1626: 73 74 jae 0x169c

1628: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

162b: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

162e: 73 65 jae 0x1695

1630: 74 3c je 0x166e

1632: 31 36 xor DWORD PTR [esi],esi

1634: 3e 28 22 sub BYTE PTR ds:[edx],ah

1637: 30 30 xor BYTE PTR [eax],dh

1639: 30 30 xor BYTE PTR [eax],dh

163b: 30 30 xor BYTE PTR [eax],dh

163d: 30 30 xor BYTE PTR [eax],dh

163f: 30 30 xor BYTE PTR [eax],dh

1641: 30 30 xor BYTE PTR [eax],dh

1643: 30 31 xor BYTE PTR [ecx],dh

1645: 30 31 xor BYTE PTR [ecx],dh

1647: 22 29 and ch,BYTE PTR [ecx]

1649: 29 3b sub DWORD PTR [ebx],edi

164b: 20 2f and BYTE PTR [edi],ch

164d: 2f das

164e: 20 4d 4f and BYTE PTR [ebp+0x4f],cl

1651: 56 push esi

1652: 20 35 20 74 6f 20 and BYTE PTR ds:0x206f7420,dh

1658: 72 65 jb 0x16bf

165a: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x31207265

1661: 31

1662: 0a 20 or ah,BYTE PTR [eax]

1664: 20 20 and BYTE PTR [eax],ah

1666: 20 0a and BYTE PTR [edx],cl

1668: 20 20 and BYTE PTR [eax],ah

166a: 20 20 and BYTE PTR [eax],ah

166c: 63 70 75 arpl WORD PTR [eax+0x75],si

166f: 2e 41 cs inc ecx

1671: 44 inc esp

1672: 44 inc esp

1673: 28 30 sub BYTE PTR [eax],dh

1675: 2c 20 sub al,0x20

1677: 31 29 xor DWORD PTR [ecx],ebp

1679: 3b 20 cmp esp,DWORD PTR [eax]

167b: 2f das

167c: 2f das

167d: 20 41 44 and BYTE PTR [ecx+0x44],al

1680: 44 inc esp

1681: 20 72 65 and BYTE PTR [edx+0x65],dh

1684: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x31207265

168b: 31

168c: 20 74 6f 20 and BYTE PTR [edi+ebp\*2+0x20],dh

1690: 72 65 jb 0x16f7

1692: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x30207265

1699: 30

169a: 20 28 and BYTE PTR [eax],ch

169c: 31 30 xor DWORD PTR [eax],esi

169e: 20 2b and BYTE PTR [ebx],ch

16a0: 20 35 29 0a 20 20 and BYTE PTR ds:0x20200a29,dh

16a6: 20 20 and BYTE PTR [eax],ah

16a8: 63 70 75 arpl WORD PTR [eax+0x75],si

16ab: 2e 50 cs push eax

16ad: 55 push ebp

16ae: 53 push ebx

16af: 48 dec eax

16b0: 28 30 sub BYTE PTR [eax],dh

16b2: 29 3b sub DWORD PTR [ebx],edi

16b4: 20 20 and BYTE PTR [eax],ah

16b6: 20 2f and BYTE PTR [edi],ch

16b8: 2f das

16b9: 20 50 55 and BYTE PTR [eax+0x55],dl

16bc: 53 push ebx

16bd: 48 dec eax

16be: 20 72 65 and BYTE PTR [edx+0x65],dh

16c1: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x30207265

16c8: 30

16c9: 20 6f 6e and BYTE PTR [edi+0x6e],ch

16cc: 74 6f je 0x173d

16ce: 20 74 68 65 and BYTE PTR [eax+ebp\*2+0x65],dh

16d2: 20 73 74 and BYTE PTR [ebx+0x74],dh

16d5: 61 popa

16d6: 63 6b 0a arpl WORD PTR [ebx+0xa],bp

16d9: 20 20 and BYTE PTR [eax],ah

16db: 20 20 and BYTE PTR [eax],ah

16dd: 63 70 75 arpl WORD PTR [eax+0x75],si

16e0: 2e 50 cs push eax

16e2: 4f dec edi

16e3: 50 push eax

16e4: 28 32 sub BYTE PTR [edx],dh

16e6: 29 3b sub DWORD PTR [ebx],edi

16e8: 20 20 and BYTE PTR [eax],ah

16ea: 20 20 and BYTE PTR [eax],ah

16ec: 2f das

16ed: 2f das

16ee: 20 50 4f and BYTE PTR [eax+0x4f],dl

16f1: 50 push eax

16f2: 20 74 68 65 and BYTE PTR [eax+ebp\*2+0x65],dh

16f6: 20 76 61 and BYTE PTR [esi+0x61],dh

16f9: 6c ins BYTE PTR es:[edi],dx

16fa: 75 65 jne 0x1761

16fc: 20 69 6e and BYTE PTR [ecx+0x6e],ch

16ff: 74 6f je 0x1770

1701: 20 72 65 and BYTE PTR [edx+0x65],dh

1704: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x32207265

170b: 32

170c: 0a 20 or ah,BYTE PTR [eax]

170e: 20 20 and BYTE PTR [eax],ah

1710: 20 0a and BYTE PTR [edx],cl

1712: 20 20 and BYTE PTR [eax],ah

1714: 20 20 and BYTE PTR [eax],ah

1716: 63 70 75 arpl WORD PTR [eax+0x75],si

1719: 2e 53 cs push ebx

171b: 55 push ebp

171c: 42 inc edx

171d: 28 32 sub BYTE PTR [edx],dh

171f: 2c 20 sub al,0x20

1721: 31 29 xor DWORD PTR [ecx],ebp

1723: 3b 20 cmp esp,DWORD PTR [eax]

1725: 2f das

1726: 2f das

1727: 20 53 55 and BYTE PTR [ebx+0x55],dl

172a: 42 inc edx

172b: 20 72 65 and BYTE PTR [edx+0x65],dh

172e: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x31207265

1735: 31

1736: 20 66 72 and BYTE PTR [esi+0x72],ah

1739: 6f outs dx,DWORD PTR ds:[esi]

173a: 6d ins DWORD PTR es:[edi],dx

173b: 20 72 65 and BYTE PTR [edx+0x65],dh

173e: 67 69 73 74 65 72 20 imul esi,DWORD PTR [bp+di+0x74],0x32207265

1745: 32

1746: 20 28 and BYTE PTR [eax],ch

1748: 72 65 jb 0x17af

174a: 73 75 jae 0x17c1

174c: 6c ins BYTE PTR es:[edi],dx

174d: 74 20 je 0x176f

174f: 73 68 jae 0x17b9

1751: 6f outs dx,DWORD PTR ds:[esi]

1752: 75 6c jne 0x17c0

1754: 64 20 62 65 and BYTE PTR fs:[edx+0x65],ah

1758: 20 30 and BYTE PTR [eax],dh

175a: 29 0a sub DWORD PTR [edx],ecx

175c: 20 20 and BYTE PTR [eax],ah

175e: 20 20 and BYTE PTR [eax],ah

1760: 0a 20 or ah,BYTE PTR [eax]

1762: 20 20 and BYTE PTR [eax],ah

1764: 20 2f and BYTE PTR [edi],ch

1766: 2f das

1767: 20 54 65 73 and BYTE PTR [ebp+eiz\*2+0x73],dl

176b: 74 20 je 0x178d

176d: 61 popa

176e: 74 6f je 0x17df

1770: 6d ins DWORD PTR es:[edi],dx

1771: 69 63 20 65 78 63 68 imul esp,DWORD PTR [ebx+0x20],0x68637865

1778: 61 popa

1779: 6e outs dx,BYTE PTR ds:[esi]

177a: 67 65 20 61 6e and BYTE PTR gs:[bx+di+0x6e],ah

177f: 64 20 63 6f and BYTE PTR fs:[ebx+0x6f],ah

1783: 6d ins DWORD PTR es:[edi],dx

1784: 70 61 jo 0x17e7

1786: 72 65 jb 0x17ed

1788: 20 65 78 and BYTE PTR [ebp+0x78],ah

178b: 63 68 61 arpl WORD PTR [eax+0x61],bp

178e: 6e outs dx,BYTE PTR ds:[esi]

178f: 67 65 0a 20 or ah,BYTE PTR gs:[bx+si]

1793: 20 20 and BYTE PTR [eax],ah

1795: 20 63 70 and BYTE PTR [ebx+0x70],ah

1798: 75 2e jne 0x17c8

179a: 58 pop eax

179b: 43 inc ebx

179c: 48 dec eax

179d: 47 inc edi

179e: 28 30 sub BYTE PTR [eax],dh

17a0: 2c 20 sub al,0x20

17a2: 31 29 xor DWORD PTR [ecx],ebp

17a4: 3b 0a cmp ecx,DWORD PTR [edx]

17a6: 20 20 and BYTE PTR [eax],ah

17a8: 20 20 and BYTE PTR [eax],ah

17aa: 63 70 75 arpl WORD PTR [eax+0x75],si

17ad: 2e 43 cs inc ebx

17af: 4d dec ebp

17b0: 50 push eax

17b1: 58 pop eax

17b2: 43 inc ebx

17b3: 48 dec eax

17b4: 47 inc edi

17b5: 28 30 sub BYTE PTR [eax],dh

17b7: 2c 20 sub al,0x20

17b9: 31 29 xor DWORD PTR [ecx],ebp

17bb: 3b 0a cmp ecx,DWORD PTR [edx]

17bd: 7d 0a jge 0x17c9

17bf: 0a 2f or ch,BYTE PTR [edi]

17c1: 2f das

17c2: 20 4d 61 and BYTE PTR [ebp+0x61],cl

17c5: 69 6e 20 66 75 6e 63 imul ebp,DWORD PTR [esi+0x20],0x636e7566

17cc: 74 69 je 0x1837

17ce: 6f outs dx,DWORD PTR ds:[esi]

17cf: 6e outs dx,BYTE PTR ds:[esi]

17d0: 0a 69 6e or ch,BYTE PTR [ecx+0x6e]

17d3: 74 20 je 0x17f5

17d5: 6d ins DWORD PTR es:[edi],dx

17d6: 61 popa

17d7: 69 6e 28 29 20 7b 0a imul ebp,DWORD PTR [esi+0x28],0xa7b2029

17de: 20 20 and BYTE PTR [eax],ah

17e0: 20 20 and BYTE PTR [eax],ah

17e2: 43 inc ebx

17e3: 50 push eax

17e4: 55 push ebp

17e5: 20 63 6f and BYTE PTR [ebx+0x6f],ah

17e8: 72 65 jb 0x184f

17ea: 31 28 xor DWORD PTR [eax],ebp

17ec: 31 30 xor DWORD PTR [eax],esi

17ee: 32 34 29 xor dh,BYTE PTR [ecx+ebp\*1]

17f1: 2c 20 sub al,0x20

17f3: 63 6f 72 arpl WORD PTR [edi+0x72],bp

17f6: 65 32 28 xor ch,BYTE PTR gs:[eax]

17f9: 31 30 xor DWORD PTR [eax],esi

17fb: 32 34 29 xor dh,BYTE PTR [ecx+ebp\*1]

17fe: 3b 0a cmp ecx,DWORD PTR [edx]

1800: 20 20 and BYTE PTR [eax],ah

1802: 20 20 and BYTE PTR [eax],ah

1804: 0a 20 or ah,BYTE PTR [eax]

1806: 20 20 and BYTE PTR [eax],ah

1808: 20 73 74 and BYTE PTR [ebx+0x74],dh

180b: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

180e: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

1811: 73 65 jae 0x1878

1813: 74 3c je 0x1851

1815: 31 36 xor DWORD PTR [esi],esi

1817: 3e 20 41 28 and BYTE PTR ds:[ecx+0x28],al

181b: 22 30 and dh,BYTE PTR [eax]

181d: 30 30 xor BYTE PTR [eax],dh

181f: 30 30 xor BYTE PTR [eax],dh

1821: 30 30 xor BYTE PTR [eax],dh

1823: 30 30 xor BYTE PTR [eax],dh

1825: 30 30 xor BYTE PTR [eax],dh

1827: 30 31 xor BYTE PTR [ecx],dh

1829: 30 31 xor BYTE PTR [ecx],dh

182b: 30 22 xor BYTE PTR [edx],ah

182d: 29 3b sub DWORD PTR [ebx],edi

182f: 20 2f and BYTE PTR [edi],ch

1831: 2f das

1832: 20 31 and BYTE PTR [ecx],dh

1834: 30 0a xor BYTE PTR [edx],cl

1836: 20 20 and BYTE PTR [eax],ah

1838: 20 20 and BYTE PTR [eax],ah

183a: 73 74 jae 0x18b0

183c: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

183f: 62 69 74 bound ebp,QWORD PTR [ecx+0x74]

1842: 73 65 jae 0x18a9

1844: 74 3c je 0x1882

1846: 31 36 xor DWORD PTR [esi],esi

1848: 3e 20 42 28 and BYTE PTR ds:[edx+0x28],al

184c: 22 30 and dh,BYTE PTR [eax]

184e: 30 30 xor BYTE PTR [eax],dh

1850: 30 30 xor BYTE PTR [eax],dh

1852: 30 30 xor BYTE PTR [eax],dh

1854: 30 30 xor BYTE PTR [eax],dh

1856: 30 30 xor BYTE PTR [eax],dh

1858: 30 30 xor BYTE PTR [eax],dh

185a: 31 30 xor DWORD PTR [eax],esi

185c: 31 22 xor DWORD PTR [edx],esp

185e: 29 3b sub DWORD PTR [ebx],edi

1860: 20 2f and BYTE PTR [edi],ch

1862: 2f das

1863: 20 35 0a 20 20 20 and BYTE PTR ds:0x2020200a,dh

1869: 20 0a and BYTE PTR [edx],cl

186b: 20 20 and BYTE PTR [eax],ah

186d: 20 20 and BYTE PTR [eax],ah

186f: 2f das

1870: 2f das

1871: 20 41 4c and BYTE PTR [ecx+0x4c],al

1874: 55 push ebp

1875: 20 6f 70 and BYTE PTR [edi+0x70],ch

1878: 65 72 61 gs jb 0x18dc

187b: 74 69 je 0x18e6

187d: 6f outs dx,DWORD PTR ds:[esi]

187e: 6e outs dx,BYTE PTR ds:[esi]

187f: 73 0a jae 0x188b

1881: 20 20 and BYTE PTR [eax],ah

1883: 20 20 and BYTE PTR [eax],ah

1885: 73 74 jae 0x18fb

1887: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

188a: 63 6f 75 arpl WORD PTR [edi+0x75],bp

188d: 74 20 je 0x18af

188f: 3c 3c cmp al,0x3c

1891: 20 22 and BYTE PTR [edx],ah

1893: 41 inc ecx

1894: 20 2b and BYTE PTR [ebx],ch

1896: 20 42 20 and BYTE PTR [edx+0x20],al

1899: 3d 20 22 20 3c cmp eax,0x3c202220

189e: 3c 20 cmp al,0x20

18a0: 41 inc ecx

18a1: 4c dec esp

18a2: 55 push ebp

18a3: 28 41 2c sub BYTE PTR [ecx+0x2c],al

18a6: 20 42 2c and BYTE PTR [edx+0x2c],al

18a9: 20 66 61 and BYTE PTR [esi+0x61],ah

18ac: 6c ins BYTE PTR es:[edi],dx

18ad: 73 65 jae 0x1914

18af: 2c 20 sub al,0x20

18b1: 66 61 popaw

18b3: 6c ins BYTE PTR es:[edi],dx

18b4: 73 65 jae 0x191b

18b6: 2c 20 sub al,0x20

18b8: 66 61 popaw

18ba: 6c ins BYTE PTR es:[edi],dx

18bb: 73 65 jae 0x1922

18bd: 29 20 sub DWORD PTR [eax],esp

18bf: 3c 3c cmp al,0x3c

18c1: 20 22 and BYTE PTR [edx],ah

18c3: 5c pop esp

18c4: 6e outs dx,BYTE PTR ds:[esi]

18c5: 22 3b and bh,BYTE PTR [ebx]

18c7: 0a 20 or ah,BYTE PTR [eax]

18c9: 20 20 and BYTE PTR [eax],ah

18cb: 20 73 74 and BYTE PTR [ebx+0x74],dh

18ce: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

18d1: 63 6f 75 arpl WORD PTR [edi+0x75],bp

18d4: 74 20 je 0x18f6

18d6: 3c 3c cmp al,0x3c

18d8: 20 22 and BYTE PTR [edx],ah

18da: 41 inc ecx

18db: 20 2d 20 42 20 3d and BYTE PTR ds:0x3d204220,ch

18e1: 20 22 and BYTE PTR [edx],ah

18e3: 20 3c 3c and BYTE PTR [esp+edi\*1],bh

18e6: 20 41 4c and BYTE PTR [ecx+0x4c],al

18e9: 55 push ebp

18ea: 28 41 2c sub BYTE PTR [ecx+0x2c],al

18ed: 20 42 2c and BYTE PTR [edx+0x2c],al

18f0: 20 74 72 75 and BYTE PTR [edx+esi\*2+0x75],dh

18f4: 65 2c 20 gs sub al,0x20

18f7: 66 61 popaw

18f9: 6c ins BYTE PTR es:[edi],dx

18fa: 73 65 jae 0x1961

18fc: 2c 20 sub al,0x20

18fe: 66 61 popaw

1900: 6c ins BYTE PTR es:[edi],dx

1901: 73 65 jae 0x1968

1903: 29 20 sub DWORD PTR [eax],esp

1905: 3c 3c cmp al,0x3c

1907: 20 22 and BYTE PTR [edx],ah

1909: 5c pop esp

190a: 6e outs dx,BYTE PTR ds:[esi]

190b: 22 3b and bh,BYTE PTR [ebx]

190d: 0a 20 or ah,BYTE PTR [eax]

190f: 20 20 and BYTE PTR [eax],ah

1911: 20 73 74 and BYTE PTR [ebx+0x74],dh

1914: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1917: 63 6f 75 arpl WORD PTR [edi+0x75],bp

191a: 74 20 je 0x193c

191c: 3c 3c cmp al,0x3c

191e: 20 22 and BYTE PTR [edx],ah

1920: 41 inc ecx

1921: 20 2a and BYTE PTR [edx],ch

1923: 20 42 20 and BYTE PTR [edx+0x20],al

1926: 3d 20 22 20 3c cmp eax,0x3c202220

192b: 3c 20 cmp al,0x20

192d: 41 inc ecx

192e: 4c dec esp

192f: 55 push ebp

1930: 28 41 2c sub BYTE PTR [ecx+0x2c],al

1933: 20 42 2c and BYTE PTR [edx+0x2c],al

1936: 20 66 61 and BYTE PTR [esi+0x61],ah

1939: 6c ins BYTE PTR es:[edi],dx

193a: 73 65 jae 0x19a1

193c: 2c 20 sub al,0x20

193e: 74 72 je 0x19b2

1940: 75 65 jne 0x19a7

1942: 2c 20 sub al,0x20

1944: 66 61 popaw

1946: 6c ins BYTE PTR es:[edi],dx

1947: 73 65 jae 0x19ae

1949: 29 20 sub DWORD PTR [eax],esp

194b: 3c 3c cmp al,0x3c

194d: 20 22 and BYTE PTR [edx],ah

194f: 5c pop esp

1950: 6e outs dx,BYTE PTR ds:[esi]

1951: 22 3b and bh,BYTE PTR [ebx]

1953: 0a 20 or ah,BYTE PTR [eax]

1955: 20 20 and BYTE PTR [eax],ah

1957: 20 73 74 and BYTE PTR [ebx+0x74],dh

195a: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

195d: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1960: 74 20 je 0x1982

1962: 3c 3c cmp al,0x3c

1964: 20 22 and BYTE PTR [edx],ah

1966: 41 inc ecx

1967: 20 2f and BYTE PTR [edi],ch

1969: 20 42 20 and BYTE PTR [edx+0x20],al

196c: 3d 20 22 20 3c cmp eax,0x3c202220

1971: 3c 20 cmp al,0x20

1973: 41 inc ecx

1974: 4c dec esp

1975: 55 push ebp

1976: 28 41 2c sub BYTE PTR [ecx+0x2c],al

1979: 20 42 2c and BYTE PTR [edx+0x2c],al

197c: 20 66 61 and BYTE PTR [esi+0x61],ah

197f: 6c ins BYTE PTR es:[edi],dx

1980: 73 65 jae 0x19e7

1982: 2c 20 sub al,0x20

1984: 66 61 popaw

1986: 6c ins BYTE PTR es:[edi],dx

1987: 73 65 jae 0x19ee

1989: 2c 20 sub al,0x20

198b: 74 72 je 0x19ff

198d: 75 65 jne 0x19f4

198f: 29 20 sub DWORD PTR [eax],esp

1991: 3c 3c cmp al,0x3c

1993: 20 22 and BYTE PTR [edx],ah

1995: 5c pop esp

1996: 6e outs dx,BYTE PTR ds:[esi]

1997: 22 3b and bh,BYTE PTR [ebx]

1999: 0a 20 or ah,BYTE PTR [eax]

199b: 20 20 and BYTE PTR [eax],ah

199d: 20 73 74 and BYTE PTR [ebx+0x74],dh

19a0: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

19a3: 63 6f 75 arpl WORD PTR [edi+0x75],bp

19a6: 74 20 je 0x19c8

19a8: 3c 3c cmp al,0x3c

19aa: 20 22 and BYTE PTR [edx],ah

19ac: 49 dec ecx

19ad: 4e dec esi

19ae: 43 inc ebx

19af: 28 41 29 sub BYTE PTR [ecx+0x29],al

19b2: 20 3d 20 22 20 3c and BYTE PTR ds:0x3c202220,bh

19b8: 3c 20 cmp al,0x20

19ba: 49 dec ecx

19bb: 4e dec esi

19bc: 43 inc ebx

19bd: 28 41 29 sub BYTE PTR [ecx+0x29],al

19c0: 20 3c 3c and BYTE PTR [esp+edi\*1],bh

19c3: 20 22 and BYTE PTR [edx],ah

19c5: 5c pop esp

19c6: 6e outs dx,BYTE PTR ds:[esi]

19c7: 22 3b and bh,BYTE PTR [ebx]

19c9: 0a 20 or ah,BYTE PTR [eax]

19cb: 20 20 and BYTE PTR [eax],ah

19cd: 20 73 74 and BYTE PTR [ebx+0x74],dh

19d0: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

19d3: 63 6f 75 arpl WORD PTR [edi+0x75],bp

19d6: 74 20 je 0x19f8

19d8: 3c 3c cmp al,0x3c

19da: 20 22 and BYTE PTR [edx],ah

19dc: 44 inc esp

19dd: 45 inc ebp

19de: 43 inc ebx

19df: 28 41 29 sub BYTE PTR [ecx+0x29],al

19e2: 20 3d 20 22 20 3c and BYTE PTR ds:0x3c202220,bh

19e8: 3c 20 cmp al,0x20

19ea: 44 inc esp

19eb: 45 inc ebp

19ec: 43 inc ebx

19ed: 28 41 29 sub BYTE PTR [ecx+0x29],al

19f0: 20 3c 3c and BYTE PTR [esp+edi\*1],bh

19f3: 20 22 and BYTE PTR [edx],ah

19f5: 5c pop esp

19f6: 6e outs dx,BYTE PTR ds:[esi]

19f7: 22 3b and bh,BYTE PTR [ebx]

19f9: 0a 20 or ah,BYTE PTR [eax]

19fb: 20 20 and BYTE PTR [eax],ah

19fd: 20 0a and BYTE PTR [edx],cl

19ff: 20 20 and BYTE PTR [eax],ah

1a01: 20 20 and BYTE PTR [eax],ah

1a03: 2f das

1a04: 2f das

1a05: 20 53 68 and BYTE PTR [ebx+0x68],dl

1a08: 69 66 74 20 6f 70 65 imul esp,DWORD PTR [esi+0x74],0x65706f20

1a0f: 72 61 jb 0x1a72

1a11: 74 69 je 0x1a7c

1a13: 6f outs dx,DWORD PTR ds:[esi]

1a14: 6e outs dx,BYTE PTR ds:[esi]

1a15: 73 0a jae 0x1a21

1a17: 20 20 and BYTE PTR [eax],ah

1a19: 20 20 and BYTE PTR [eax],ah

1a1b: 73 74 jae 0x1a91

1a1d: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1a20: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1a23: 74 20 je 0x1a45

1a25: 3c 3c cmp al,0x3c

1a27: 20 22 and BYTE PTR [edx],ah

1a29: 53 push ebx

1a2a: 48 dec eax

1a2b: 4c dec esp

1a2c: 28 41 29 sub BYTE PTR [ecx+0x29],al

1a2f: 20 3d 20 22 20 3c and BYTE PTR ds:0x3c202220,bh

1a35: 3c 20 cmp al,0x20

1a37: 53 push ebx

1a38: 48 dec eax

1a39: 4c dec esp

1a3a: 28 41 29 sub BYTE PTR [ecx+0x29],al

1a3d: 20 3c 3c and BYTE PTR [esp+edi\*1],bh

1a40: 20 22 and BYTE PTR [edx],ah

1a42: 5c pop esp

1a43: 6e outs dx,BYTE PTR ds:[esi]

1a44: 22 3b and bh,BYTE PTR [ebx]

1a46: 0a 20 or ah,BYTE PTR [eax]

1a48: 20 20 and BYTE PTR [eax],ah

1a4a: 20 73 74 and BYTE PTR [ebx+0x74],dh

1a4d: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1a50: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1a53: 74 20 je 0x1a75

1a55: 3c 3c cmp al,0x3c

1a57: 20 22 and BYTE PTR [edx],ah

1a59: 53 push ebx

1a5a: 48 dec eax

1a5b: 52 push edx

1a5c: 28 41 29 sub BYTE PTR [ecx+0x29],al

1a5f: 20 3d 20 22 20 3c and BYTE PTR ds:0x3c202220,bh

1a65: 3c 20 cmp al,0x20

1a67: 53 push ebx

1a68: 48 dec eax

1a69: 52 push edx

1a6a: 28 41 29 sub BYTE PTR [ecx+0x29],al

1a6d: 20 3c 3c and BYTE PTR [esp+edi\*1],bh

1a70: 20 22 and BYTE PTR [edx],ah

1a72: 5c pop esp

1a73: 6e outs dx,BYTE PTR ds:[esi]

1a74: 22 3b and bh,BYTE PTR [ebx]

1a76: 0a 20 or ah,BYTE PTR [eax]

1a78: 20 20 and BYTE PTR [eax],ah

1a7a: 20 0a and BYTE PTR [edx],cl

1a7c: 20 20 and BYTE PTR [eax],ah

1a7e: 20 20 and BYTE PTR [eax],ah

1a80: 2f das

1a81: 2f das

1a82: 20 4d 75 and BYTE PTR [ebp+0x75],cl

1a85: 6c ins BYTE PTR es:[edi],dx

1a86: 74 69 je 0x1af1

1a88: 2d 63 6f 72 65 sub eax,0x65726f63

1a8d: 20 65 78 and BYTE PTR [ebp+0x78],ah

1a90: 65 63 75 74 arpl WORD PTR gs:[ebp+0x74],si

1a94: 69 6f 6e 0a 20 20 20 imul ebp,DWORD PTR [edi+0x6e],0x2020200a

1a9b: 20 73 74 and BYTE PTR [ebx+0x74],dh

1a9e: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1aa1: 74 68 je 0x1b0b

1aa3: 72 65 jb 0x1b0a

1aa5: 61 popa

1aa6: 64 20 74 31 28 and BYTE PTR fs:[ecx+esi\*1+0x28],dh

1aab: 65 78 65 gs js 0x1b13

1aae: 63 75 74 arpl WORD PTR [ebp+0x74],si

1ab1: 65 5f gs pop edi

1ab3: 63 6f 72 arpl WORD PTR [edi+0x72],bp

1ab6: 65 2c 20 gs sub al,0x20

1ab9: 73 74 jae 0x1b2f

1abb: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1abe: 72 65 jb 0x1b25

1ac0: 66 28 63 6f data16 sub BYTE PTR [ebx+0x6f],ah

1ac4: 72 65 jb 0x1b2b

1ac6: 31 29 xor DWORD PTR [ecx],ebp

1ac8: 2c 20 sub al,0x20

1aca: 31 29 xor DWORD PTR [ecx],ebp

1acc: 3b 0a cmp ecx,DWORD PTR [edx]

1ace: 20 20 and BYTE PTR [eax],ah

1ad0: 20 20 and BYTE PTR [eax],ah

1ad2: 73 74 jae 0x1b48

1ad4: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1ad7: 74 68 je 0x1b41

1ad9: 72 65 jb 0x1b40

1adb: 61 popa

1adc: 64 20 74 32 28 and BYTE PTR fs:[edx+esi\*1+0x28],dh

1ae1: 65 78 65 gs js 0x1b49

1ae4: 63 75 74 arpl WORD PTR [ebp+0x74],si

1ae7: 65 5f gs pop edi

1ae9: 63 6f 72 arpl WORD PTR [edi+0x72],bp

1aec: 65 2c 20 gs sub al,0x20

1aef: 73 74 jae 0x1b65

1af1: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1af4: 72 65 jb 0x1b5b

1af6: 66 28 63 6f data16 sub BYTE PTR [ebx+0x6f],ah

1afa: 72 65 jb 0x1b61

1afc: 32 29 xor ch,BYTE PTR [ecx]

1afe: 2c 20 sub al,0x20

1b00: 32 29 xor ch,BYTE PTR [ecx]

1b02: 3b 0a cmp ecx,DWORD PTR [edx]

1b04: 20 20 and BYTE PTR [eax],ah

1b06: 20 20 and BYTE PTR [eax],ah

1b08: 0a 20 or ah,BYTE PTR [eax]

1b0a: 20 20 and BYTE PTR [eax],ah

1b0c: 20 74 31 2e and BYTE PTR [ecx+esi\*1+0x2e],dh

1b10: 6a 6f push 0x6f

1b12: 69 6e 28 29 3b 0a 20 imul ebp,DWORD PTR [esi+0x28],0x200a3b29

1b19: 20 20 and BYTE PTR [eax],ah

1b1b: 20 74 32 2e and BYTE PTR [edx+esi\*1+0x2e],dh

1b1f: 6a 6f push 0x6f

1b21: 69 6e 28 29 3b 0a 0a imul ebp,DWORD PTR [esi+0x28],0xa0a3b29

1b28: 20 20 and BYTE PTR [eax],ah

1b2a: 20 20 and BYTE PTR [eax],ah

1b2c: 2f das

1b2d: 2f das

1b2e: 20 45 78 and BYTE PTR [ebp+0x78],al

1b31: 61 popa

1b32: 6d ins DWORD PTR es:[edi],dx

1b33: 70 6c jo 0x1ba1

1b35: 65 20 6f 66 and BYTE PTR gs:[edi+0x66],ch

1b39: 20 75 73 and BYTE PTR [ebp+0x73],dh

1b3c: 69 6e 67 20 63 6f 6e imul ebp,DWORD PTR [esi+0x67],0x6e6f6320

1b43: 64 69 74 69 6f 6e 61 imul esi,DWORD PTR fs:[ecx+ebp\*2+0x6f],0x206c616e

1b4a: 6c 20

1b4c: 6a 75 push 0x75

1b4e: 6d ins DWORD PTR es:[edi],dx

1b4f: 70 20 jo 0x1b71

1b51: 28 4a 5a sub BYTE PTR [edx+0x5a],cl

1b54: 20 61 6e and BYTE PTR [ecx+0x6e],ah

1b57: 64 20 4a 43 and BYTE PTR fs:[edx+0x43],cl

1b5b: 29 0a sub DWORD PTR [edx],ecx

1b5d: 20 20 and BYTE PTR [eax],ah

1b5f: 20 20 and BYTE PTR [eax],ah

1b61: 69 66 20 28 63 6f 72 imul esp,DWORD PTR [esi+0x20],0x726f6328

1b68: 65 31 2e xor DWORD PTR gs:[esi],ebp

1b6b: 4a dec edx

1b6c: 5a pop edx

1b6d: 28 29 sub BYTE PTR [ecx],ch

1b6f: 29 20 sub DWORD PTR [eax],esp

1b71: 7b 0a jnp 0x1b7d

1b73: 20 20 and BYTE PTR [eax],ah

1b75: 20 20 and BYTE PTR [eax],ah

1b77: 20 20 and BYTE PTR [eax],ah

1b79: 20 20 and BYTE PTR [eax],ah

1b7b: 73 74 jae 0x1bf1

1b7d: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1b80: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1b83: 74 20 je 0x1ba5

1b85: 3c 3c cmp al,0x3c

1b87: 20 22 and BYTE PTR [edx],ah

1b89: 4a dec edx

1b8a: 75 6d jne 0x1bf9

1b8c: 70 20 jo 0x1bae

1b8e: 74 6f je 0x1bff

1b90: 20 61 64 and BYTE PTR [ecx+0x64],ah

1b93: 64 72 65 fs jb 0x1bfb

1b96: 73 73 jae 0x1c0b

1b98: 20 28 and BYTE PTR [eax],ch

1b9a: 5a pop edx

1b9b: 65 72 6f gs jb 0x1c0d

1b9e: 20 66 6c and BYTE PTR [esi+0x6c],ah

1ba1: 61 popa

1ba2: 67 20 69 73 and BYTE PTR [bx+di+0x73],ch

1ba6: 20 73 65 and BYTE PTR [ebx+0x65],dh

1ba9: 74 29 je 0x1bd4

1bab: 5c pop esp

1bac: 6e outs dx,BYTE PTR ds:[esi]

1bad: 22 3b and bh,BYTE PTR [ebx]

1baf: 0a 20 or ah,BYTE PTR [eax]

1bb1: 20 20 and BYTE PTR [eax],ah

1bb3: 20 7d 0a and BYTE PTR [ebp+0xa],bh

1bb6: 0a 20 or ah,BYTE PTR [eax]

1bb8: 20 20 and BYTE PTR [eax],ah

1bba: 20 69 66 and BYTE PTR [ecx+0x66],ch

1bbd: 20 28 and BYTE PTR [eax],ch

1bbf: 63 6f 72 arpl WORD PTR [edi+0x72],bp

1bc2: 65 31 2e xor DWORD PTR gs:[esi],ebp

1bc5: 4a dec edx

1bc6: 43 inc ebx

1bc7: 28 29 sub BYTE PTR [ecx],ch

1bc9: 29 20 sub DWORD PTR [eax],esp

1bcb: 7b 0a jnp 0x1bd7

1bcd: 20 20 and BYTE PTR [eax],ah

1bcf: 20 20 and BYTE PTR [eax],ah

1bd1: 20 20 and BYTE PTR [eax],ah

1bd3: 20 20 and BYTE PTR [eax],ah

1bd5: 73 74 jae 0x1c4b

1bd7: 64 3a 3a cmp bh,BYTE PTR fs:[edx]

1bda: 63 6f 75 arpl WORD PTR [edi+0x75],bp

1bdd: 74 20 je 0x1bff

1bdf: 3c 3c cmp al,0x3c

1be1: 20 22 and BYTE PTR [edx],ah

1be3: 4a dec edx

1be4: 75 6d jne 0x1c53

1be6: 70 20 jo 0x1c08

1be8: 74 6f je 0x1c59

1bea: 20 61 64 and BYTE PTR [ecx+0x64],ah

1bed: 64 72 65 fs jb 0x1c55

1bf0: 73 73 jae 0x1c65

1bf2: 20 28 and BYTE PTR [eax],ch

1bf4: 43 inc ebx

1bf5: 61 popa

1bf6: 72 72 jb 0x1c6a

1bf8: 79 20 jns 0x1c1a

1bfa: 66 6c data16 ins BYTE PTR es:[edi],dx

1bfc: 61 popa

1bfd: 67 20 69 73 and BYTE PTR [bx+di+0x73],ch

1c01: 20 73 65 and BYTE PTR [ebx+0x65],dh

1c04: 74 29 je 0x1c2f

1c06: 5c pop esp

1c07: 6e outs dx,BYTE PTR ds:[esi]

1c08: 22 3b and bh,BYTE PTR [ebx]

1c0a: 0a 20 or ah,BYTE PTR [eax]

1c0c: 20 20 and BYTE PTR [eax],ah

1c0e: 20 7d 0a and BYTE PTR [ebp+0xa],bh

1c11: 0a 20 or ah,BYTE PTR [eax]

1c13: 20 20 and BYTE PTR [eax],ah

1c15: 20 72 65 and BYTE PTR [edx+0x65],dh

1c18: 74 75 je 0x1c8f

1c1a: 72 6e jb 0x1c8a

1c1c: 20 30 and BYTE PTR [eax],dh

1c1e: 3b 0a cmp ecx,DWORD PTR [edx]

1c20: 7d 0a jge 0x1c2c

Downloads

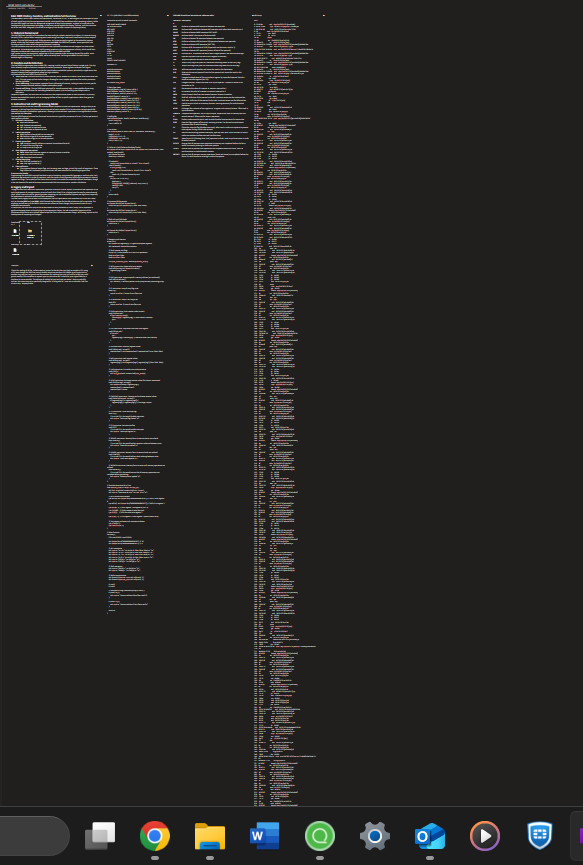
V1.1

<<intel\_3003\_x86-16\_2\_core\_emulator.cpp>><<intel-3003-emulator-main.zip>>

X86 dump

<<x86-dump.asm>>

Thankyou   
  
Thanks for reading all of this, unfortunately its quite a lot harder than you think to emulate a CPU using a CPU even though the 3003 has a much smaller instruction set than a i9-14900K its still quite hard to emulate. Any way you can go and use this emulator for anything (you just can't sell for profit) you could go and modify it for a emulator or expand upon its instruction set. It would be quite a good base for a emulator or microcontroller. The x86 was all codded by hand and took me about 2 days to code fully and test, the instruction set is specifically designed for a 16 leg x86-16 2 core micro controller, feel free to use in any . Anyway thanks.



How much of this is chatgpt?

Non

This is how you do x86 spent like 8 days