Devan Hone

Ethan Olsen

Kevin Estrada

Scrum Team 5

**Unit Test - Interpreter**

**Summary:**

A unit test was performed on the interpreter class of the UVSim software. The class was isolated from the main code and tested with a new main program which ran the interpreter through all implemented error cases and all nominal input functions. Through the testing phases a few bugs were found and corrected. The system was unsuccessfully inputting negative numbers to the memory. This was due to a double negation occurring during the parsing process. Additionally, the interpreter had the main memory size defined incorrectly and allowed writing outside of memory (into the PC and accumulator).

After corrections, the testing program was ran again and all tests passed, with the exception of one. Test #4 did not return the run-time error expected but did catch the error. This error check was decided to be redundant and acceptable.

**Program Output from all tests:**

Test #1: Input = +10105

Expected result: "invalid input: input length", Actual result: invalid input: input length...Pass

Test #2: Input = ""

Expected result: "no empty lines are allowed", Actual result: no empty lines are allowed...Pass

Test #3: Input = word

Expected result: "Incorrect argument, must be an integer", Actual result: Incorrect argument, must be an integer...Pass

Test #4: Input = ^1596

Expected result: "invalid first character", Actual result: Incorrect argument, must be an integer...Fail

Test #5: Input = ZZZZ

Expected result: 0, Actual result: 0...Pass

Test #6: Input = +1112

Expected result: 0, Actual result: 0...Pass

Test #7: Input = 3456

Expected result: 11123456, Actual result: 11123456...Pass

Test #8: Input = +1155

Expected result: 1155, Actual result: 1155...Pass

Test #9: Input = +2025

Expected result: 2025, Actual result: 2025...Pass

Test #10: Input = +2130

Expected result: 2130, Actual result: 2130...Pass

Test #11: Input = +3010

Expected result: 3010, Actual result: 3010...Pass

Test #12: Input = +3145

Expected result: 3145, Actual result: 3145...Pass

Test #13: Input = +3230

Expected result: 3230, Actual result: 3230...Pass

Test #14: Input = +3316

Expected result: 3316, Actual result: 3316...Pass

Test #15: Input = +4089

Expected result: 4089, Actual result: 4089...Pass

Test #16: Input = +4156

Expected result: 4156, Actual result: 4156...Pass

Test #17: Input = +4236

Expected result: 4236, Actual result: 4236...Pass

Test #18: Input = +4300

Expected result: 4300, Actual result: 4300...Pass

Test #19: Input = -2365

Expected result: -2365, Actual result: -2365...Pass

increasing incrementor to max memory

Test #20: Input = +1234

Expected result: "max memory exceeded, too many commands", Actual result: max memory exceeded, too many commands...Pass

Total number of tests: 20

Passed: 19, Failed: 1

PC: 0

Accumulator: 0

11123456 1155 2025 2130 3010 3145 3230 3316 4089 4156

4236 4300 -2365 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

**Unit Test Program Code:**

#include "uut\_interpreter.h"

#include <stdlib.h>

#include <string>

#include <stdio.h>

#include <iostream>

using namespace std;

int incrementor = 0;

int main(void)

{

//////////////////VARIABLES/////////////////////////

string commandToTest;

bool executionResult = false;

int cntPass = 0;

int cntFail = 0;

int numTestsCompleted = 0;

int expectedResult = 0;

int actualResult = 0;

string expectedString = "";

uut\_interpreter myInterpreter = uut\_interpreter();

vmMemory\* myMemory = vmMemory::getInstance();

////////////////////////////////////////////////////

///////////////////////////////////////10 - read//////////////////////////////////////////////

expectedString = "invalid input: input length";

try

{

cout << "Test #" << numTestsCompleted + 1 << ": Input = +10105\n";

commandToTest = "+10105";

expectedResult = 1010;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Expected result: \"invalid input: input length\"" << "failed" << endl;

cntFail++;

}

catch (const std::exception& error)

{

cout << "Expected result: \"invalid input: input length\"" << ", Actual result: " << error.what() << "...";

string actualString = error.what();

if (expectedString == actualString)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

}

numTestsCompleted++;

/////////////////////////////////////empty line///////////////////////////////////

cout << endl;

expectedString = "no empty lines are allowed";

try

{

cout << "Test #" << numTestsCompleted + 1 << ": Input = \"\"\n";

commandToTest = "";

expectedResult = 1010;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Expected result: \"no empty lines are allowed\"" << "failed" << endl;

cntFail++;

}

catch (const std::exception& error)

{

cout << "Expected result: \"no empty lines are allowed\"" << ", Actual result: " << error.what() << "...";

string actualString = error.what();

if (expectedString == actualString)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

}

numTestsCompleted++;

///////////////////////////////////not signed int/////////////////////////////////////////////

cout << endl;

expectedString = "Incorrect argument, must be an integer";

try

{

cout << "Test #" << numTestsCompleted + 1 << ": Input = word\n";

commandToTest = "word";

expectedResult = 1010;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Expected result: \"Incorrect argument, must be an integer\"" << "failed" << endl;

cntFail++;

}

catch (const std::exception& error)

{

cout << "Expected result: \"Incorrect argument, must be an integer\"" << ", Actual result: " << error.what() << "...";

string actualString = error.what();

if (expectedString == actualString)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

}

numTestsCompleted++;

////////////////////////////////////invalid first character/////////////////////////////////////////////////

cout << endl;

expectedString = "invalid first character";

try

{

cout << "Test #" << numTestsCompleted + 1 << ": Input = ^1596\n";

commandToTest = "^1596";

expectedResult = 1010;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Expected result: \"invalid first character\"" << "failed" << endl;

cntFail++;

}

catch (const std::exception& error)

{

cout << "Expected result: \"invalid first character\"" << ", Actual result: " << error.what() << "...";

string actualString = error.what();

if (expectedString == actualString)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

}

numTestsCompleted++;

//////////////////////////////////////////////zzzz input///////////////////////////////////////

cout << endl;

commandToTest = "ZZZZ";

expectedResult = 0;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = ZZZZ\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

cout << endl;

commandToTest = "+1112";

expectedResult = 0;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +1112\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

cout << endl;

commandToTest = "3456";

expectedResult = 11123456;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = 3456\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

//cout << myMemory->dump();

/////////////////////////////////////11 - write///////////////////////////////////////////////

cout << endl;

commandToTest = "+1155";

expectedResult = 1155;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +1155\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

/////////////////////////////////////////20 - load///////////////////////////////////////////////

cout << endl;

commandToTest = "+2025";

expectedResult = 2025;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +2025\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

/////////////////////////////////////////21 - store/////////////////////////////////////////////////

cout << endl;

commandToTest = "+2130";

expectedResult = 2130;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +2130\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

////////////////////////////////////////30 - add/////////////////////////////////////////////////////

cout << endl;

commandToTest = "+3010";

expectedResult = 3010;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +3010\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

//////////////////////////////////////////////31 - sub///////////////////////////////////////////

cout << endl;

commandToTest = "+3145";

expectedResult = 3145;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +3145\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

///////////////////////////////////////////////32 - div////////////////////////////////////////

cout << endl;

commandToTest = "+3230";

expectedResult = 3230;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +3230\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

///////////////////////////////////////////////33 - mult////////////////////////////////////////

cout << endl;

commandToTest = "+3316";

expectedResult = 3316;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +3316\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

//////////////////////////////////////////////40 - br/////////////////////////////////////////////

cout << endl;

commandToTest = "+4089";

expectedResult = 4089;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +4089\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

///////////////////////////////////////////////41 - brn////////////////////////////////////////

cout << endl;

commandToTest = "+4156";

expectedResult = 4156;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +4156\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

//////////////////////////////////////////42 - brz///////////////////////////////////////////

cout << endl;

commandToTest = "+4236";

expectedResult = 4236;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +4236\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

//////////////////////////////////////////43 - halt/////////////////////////////////////////

cout << endl;

commandToTest = "+4300";

expectedResult = 4300;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = +4300\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

//////////////////////////////////////// negative number input //////////////////////////////////

cout << endl;

commandToTest = "-2365";

expectedResult = -2365;

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Test #" << numTestsCompleted + 1 << ": Input = -2365\n";

cout << "Expected result: " << expectedResult << ", Actual result: " << actualResult << "...";

if (expectedResult == actualResult)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

numTestsCompleted++;

incrementor++;

///////////////////////////////////////////////MAX memory/////////////////////////////////////////////

cout << "increasing incrementor to max memory" << endl;

for (incrementor; incrementor < 100; incrementor++)

{

commandToTest = "+1111";

myInterpreter.parseCommand(commandToTest);

}

cout << endl;

expectedString = "max memory exceeded, too many commands";

try

{

cout << "Test #" << numTestsCompleted + 1 << ": Input = +1234\n";

commandToTest = "+1234";

myInterpreter.parseCommand(commandToTest);

myMemory->read(incrementor, &actualResult);

cout << "Expected result: \"max memory exceeded, too many commands\"" << "failed" << endl;

cntFail++;

}

catch (const std::exception& error)

{

cout << "Expected result: \"max memory exceeded, too many commands\"" << ", Actual result: " << error.what() << "...";

string actualString = error.what();

if (expectedString == actualString)

{

cntPass++;

cout << "Pass" << endl;

}

else

{

cntFail++;

cout << "Fail" << endl;

}

}

numTestsCompleted++;

cout << "Total number of tests: " << numTestsCompleted << endl;

cout << "Passed: " << cntPass <<", Failed: " << cntFail << endl;

cout << myMemory->dump();

return 0;

}