|  |
| --- |
|  |
| Data Dictionary for  main.c |
|  |
|  |
| **Andrew Laing** |
| **09/09/2017** |

Contents

[Using the Simple Compiler 2](#_Toc492666156)

[main () 2](#_Toc492666157)

[initialiseFlags() 2](#_Toc492666158)

[printFlags() 2](#_Toc492666159)

[printInstructions() 2](#_Toc492666160)

[writeInstructionsToFile() 3](#_Toc492666161)

[addLineNumberToTable() 3](#_Toc492666162)

[addVariableToTable () 3](#_Toc492666163)

[addConstantToTable() 4](#_Toc492666164)

[runFirstPass() 4](#_Toc492666165)

[runSecondPass() 4](#_Toc492666166)

[createTokenArray() 5](#_Toc492666167)

[addSymbolsToTable() 5](#_Toc492666168)

[createInputInstruction() 6](#_Toc492666169)

[createPrintInstruction() 6](#_Toc492666170)

[createGotoInstruction() 6](#_Toc492666171)

[createEndInstruction() 7](#_Toc492666172)

[createLetInstruction() 7](#_Toc492666173)

[createIfInstruction() 8](#_Toc492666174)

[dumpTableEntries() 8](#_Toc492666175)

[variableExistsInTable() 9](#_Toc492666176)

[constantExistsInTable() 9](#_Toc492666177)

[lineNumberExistsInTable() 9](#_Toc492666178)

[convertToPostfix () 9](#_Toc492666179)

[precedence() 10](#_Toc492666180)

[evaluatePostfixExpression() 10](#_Toc492666181)

[calculate() 10](#_Toc492666182)

# Using the Simple Compiler

To use the Simple Compiler just run it from the command line. It will ask for the name of the file to compile, which should have a ‘.simple ’ suffix. This Simple file should be located within the same directory as the compiler binary. The compiled SML file will also be written out to this directory as ‘out.sml’

|  |  |  |
| --- | --- | --- |
| **main ()** | | |
| **Name** | **Type** | **Description** |
| filename | char [] | Used to hold the name of the Simple source code entered by the user. |
| MAXLINELENGTH | 100 | Used to limit the length of the filename read in. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **initialiseFlags()** | | |
| **Name** | **Type** | **Description** |
| i | int | Used to iterate through the Flags array. |
| MEMORYSIZE |  | Used to define the size of the SML Virtual Memory space. |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **printFlags()** | | |
| **Name** | **Type** | **Description** |
| i | int | Used to iterate through the Flags array. |
| MEMORYSIZE |  | Used to define the size of the SML Virtual Memory space. |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **printInstructions()** | | |
| **Name** | **Type** | **Description** |
| i | int | Used to iterate through the SML Virtual Memory space array. |
| MEMORYSIZE |  | Used to hold the size of the SML Virtual Memory space. Defined in SimpleCompiler.h |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **writeInstructionsToFile()** | | |
| **Name** | **Type** | **Description** |
| i | int | Used to iterate through the SML Virtual Memory space array. |
| ofPtr | FILE \* | Used to hold a pointer to the output file containing the SML instructions. |
| MEMORYSIZE |  | Used to hold the size of the SML Virtual Memory space. Defined in SimpleCompiler.h |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **addLineNumberToTable()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| PreviousLineNumber | int | Used to hold the line number of the previously read Simple line to ensure that they have been numbered correctly. (Incorrect ordering will result in the program terminating with a Syntax Error.) Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **addVariableToTable ()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| VARCONSTADDRESS | int | Used to hold the next memory address, at the end of the SML Memory space, where a value can be added. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **addConstantToTable()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| VARCONSTADDRESS | int | Used to hold the next memory address, at the end of the SML Memory space, where a value can be added. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **runFirstPass()** | | |
| **Name** | **Type** | **Description** |
| line | char [] | Used to hold a line of code read in from the Simple source file |
| lineCopy | char [] | Used to hold a copy of line because when the line is tokenised into the tokenArray using strtok, this changes the original string. |
| tokenArray | char [][] | Used to hold the tokens created from the line which will be added to the Symbol Table and used to create instructions. |
| lineParsed | int | Used as a flag to determine whether lines of source code have been parsed correctly. |
| sourceFilePtr | FILE \* | Used to hold a pointer to the Simple source file being compiled into SML. |
| numberOfTokens | int | Used to hold the number of tokens created from the line of Simple source code, and added to tokenArray. |

|  |  |  |
| --- | --- | --- |
| **runSecondPass()** | | |
| **Name** | **Type** | **Description** |
| i | int | Used to iterate through the Flags array and update the SML MEMORY address with the resolved line number address for a Branch instruction. |
| address | int | Used to hold resolved address to add to the SML instruction which did not resolve it during the first pass of the compiler. |
| MEMORYSIZE |  | Used to hold the size of the SML Virtual Memory space. Defined in SimpleCompiler.h |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **createTokenArray()** | | |
| **Name** | **Type** | **Description** |
| length | int | Used to hold the number of tokens added to the tokenArray, and is the return value of this function. |
| lineCopy | char [] | Used to hold a copy of the line passed to this function because passing the original to strtok would alter it. |
| tokenPtr | char \* | Used to hold a reference to tokens created by strtok. |
| MAXLINELENGTH |  | Used to limit the length of lines which can be read in from a Simple source file. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **addSymbolsToTable()** | | |
| **Name** | **Type** | **Description** |
| number | int | Used to hold numbers, extracted from the Simple source code, to be added to the Symbol Table. |
| isLineNumber | int | Used to as a flag to process line numbers extracted from the Simple source code, to be added to the Symbol Table. |
| i | int | Used to iterate through the token array. |
| PREVIOUSLINENUMBER | int | Used to hold the line number of the previously read Simple line to ensure that they have been numbered correctly. (Incorrect ordering will result in the program terminating with a Syntax Error.) Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **createInputInstruction()** | | |
| **Name** | **Type** | **Description** |
| instruction | int | Used to hold the instruction to be written to the SML Memory space.. |
| variableAddress | int | Used to hold the address within the SML Memory space containing the value for the variable to be READ. |
| lineNumber | int | Used to hold the line number attached to the Simple Command, converted to an integer. |
| instructionAddress | int | Used to hold the address assigned to the line number within the SML Memory space where its first instruction will be stored. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| READ |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |

|  |  |  |
| --- | --- | --- |
| **createPrintInstruction()** | | |
| **Name** | **Type** | **Description** |
| instruction | int | Used to hold the instruction to be written to the SML Memory space.. |
| variableAddress | int | Used to hold the address within the SML Memory space containing the value for the variable to be written to STDOUT. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| WRITE |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |

|  |  |  |
| --- | --- | --- |
| **createGotoInstruction()** | | |
| **Name** | **Type** | **Description** |
| instruction | int | Used to hold the instruction to be written to the SML Memory space.. |
| lineNumber | int | Used to hold the line number attached to the Simple Command, converted to an integer. |
| returnLineNumber | int | Used to hold the line number of Simple code representing the line where the goto statement will jump to. |
| instructionAddress | int | Used to hold the address assigned to the line number within the SML Memory space where its first instruction will be stored. |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |
| BRANCH |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |

|  |  |  |
| --- | --- | --- |
| **createEndInstruction()** | | |
| **Name** | **Type** | **Description** |
| instruction | int | Used to the instruction to be written to the SML Memory space.. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| HALT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |

|  |  |  |
| --- | --- | --- |
| **createLetInstruction()** | | |
| **Name** | **Type** | **Description** |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| postfix | char [] | Used to hold the expression on the right-hand side of the assignment operator in the Simple command, converted from infix to postfix. |
| variableAddress | int | Used to hold the address within the SML Memory space containing where the result of the postfix evaluation will be stored.. |
| loadInstruction | int | Used to hold the LOAD instruction to be written to the SML Memory space. |
| storeInstruction | int | Used to hold the STORE instruction to be written to the SML Memory space. |
| evalResult | int | Used to hold the temporary address in the SML Memory space where the result of evaluation the expression on the right-hand side of the assignment operator will be stored. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| STORE |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |

|  |  |  |
| --- | --- | --- |
| **createIfInstruction()** | | |
| **Name** | **Type** | **Description** |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| compOpIdx | int | Used to hold the index of the comparison operator within the token array. Will be used to split the array into two expressions to evaluate then compare. |
| LTOPERATOR |  | An item from the ComparisonOperators enum. |
| GTOPERATOR |  | An item from the ComparisonOperators enum. |
| LTEOPERATOR |  | An item from the ComparisonOperators enum. |
| GTEOPERATOR |  | An item from the ComparisonOperators enum. |
| EQOPERATOR |  | An item from the ComparisonOperators enum. |
| NEOPERATOR |  | An item from the ComparisonOperators enum. |

|  |  |  |
| --- | --- | --- |
| **dumpTableEntries()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| i | int | Used to iterate through the Symbol table. |

|  |  |  |
| --- | --- | --- |
| **variableExistsInTable()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| i | int | Used to iterate through the Symbol table. |

|  |  |  |
| --- | --- | --- |
| **constantExistsInTable()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| i | int | Used to iterate through the Symbol table. |

|  |  |  |
| --- | --- | --- |
| **lineNumberExistsInTable()** | | |
| **Name** | **Type** | **Description** |
| Table | struct tableEntry [] | An array of tableEntries. I.e., the Symbol table. Defined in SimpleCompiler.h |
| TABLEPOSITION | int | Used to hold the next available position in the Symbol Table which can accept an entry. Defined in SimpleCompiler.h |
| i | int | Used to iterate through the Symbol table. |

|  |  |  |
| --- | --- | --- |
| **convertToPostfix ()** | | |
| **Name** | **Type** | **Description** |
| i | int | Used to iterate through the infix array. |
| j | int | Used to add characters to the postfix array |
| k | int | Used to iterate through the infix array. |
| c | char | Used to hold characters from the infix array to be pushed to the stack |
| stackPtr | StackNodePtr | Used to create the postfix expression. Defined in IntStack.h |

|  |  |  |
| --- | --- | --- |
| **precedence()** | | |
| **Name** | **Type** | **Description** |
| result | int | Used to hold the result of the comparison operation which determines the operator with the higher precedence, and is the retrurn value of this function. |

|  |  |  |
| --- | --- | --- |
| **evaluatePostfixExpression()** | | |
| **Name** | **Type** | **Description** |
| stackPtr | StackNodePtr | Used when evaluating the postfix expression. Defined in IntStack.h |
| postfix | char [][] | Used to hold the postfix expression array extracted from the postfix expression string. |
| pfLength | int | Used to hold the number of items in the postfix array. |
| x | int | Used to hold the address of a variable/constant SML Memory space address popped from the stack to create an instruction with. |
| y | int | Used to hold a variable/constant SML Memory space address popped from the stack to create an instruction with. |
| i | int | Used to iterate through the postfix array. |
| address | int | Used to hold a variable/constant SML Memory space address to push to the stack. |
| c | char | Used to determine what the token in the postfix array represents, eg variable, operator… |

|  |  |  |
| --- | --- | --- |
| **calculate()** | | |
| **Name** | **Type** | **Description** |
| tempStorage | int | Used to hold an address in the SML Memory space where the result of a calculation will be stored. |
| VARCONSTADDRESS | int | Used to hold the next memory address, at the end of the SML Memory space, where a value can be added. Defined in SimpleCompiler.h |
| loadInstruction | int | Used to create a LOAD instruction to load the first operand of an expression to evaluate into the Accumulator. |
| storeInstruction | int | Used to create a STORE instruction to store the result operand of the calculation into the tempStorage address. |
| addInstruction | int | Used to create an ADD instruction. |
| subInstruction | int | Used to create a SUBTRACT instruction. |
| mulInstruction | int | Used to create a MULTIPLY instruction. |
| divInstruction | int | Used to create a DIVIDE instruction. |
| expInstruction | int | Used to create an EXPONENTIATION instruction. |
| modInstruction | int | Used to create a MODULUS instruction. |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| STORE |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| ADD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| DIVIDE |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| MULTIPLY |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| EXPONENTIATION |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| MODULUS |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **getComparisonOperatorIndex()** | | |
| **Name** | **Type** | **Description** |
| idx | int | Used to get the index of the comparison operator within the token array containing the tokens from a Simple IF statement. |
| idx2 | int | Used to iterate through the compOperators array. |
| compOperators | const char [][] | Used to hold string representations of Simple’s comparison operators. Defined in SimpleCompiler.h. |
| NUMBEROFCOMPOPERATORS | 6 | Used to define the maximum number of comparison operators allowed within the Simple language. Defined in SimpleCompiler.h. |

|  |  |  |
| --- | --- | --- |
| **isComparisonOperator()** | | |
| **Name** | **Type** | **Description** |
| idx | int | Used to get the index of the comparison operator within the compOperators array. |
| compOperators | const char [][] | Used to hold string representations of Simple’s comparison operators. Defined in SimpleCompiler.h. |
| NUMBEROFCOMPOPERATORS | 6 | Used to define the maximum number of comparison operators allowed within the Simple language. Defined in SimpleCompiler.h. |

|  |  |  |
| --- | --- | --- |
| **doLTJMP()** | | |
| **Name** | **Type** | **Description** |
| LHSAddress | int | Used to hold the address of the result of the expression on the left-hand side of the comparison operator. |
| RHSAddress | int | Used to hold the address of the result of the expression on the right-hand side of the comparison operator. |
| LHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the left-hand side of the comparison operator. |
| RHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the right-hand side of the comparison operator. |
| MAXLINELENGTH |  | Used to limit the length of the postfix expression arrays. Defined in SimpleCompiler.h |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| instruction1 | int | Used to create a LOAD instruction to load the first value of the comparison into the Accumulator. |
| instruction2 | int | Used to create a SUBTRACT instruction to second from the value stored in the Accumulator. |
| instruction3 | int | Used to create a BRANCHNEG instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is negative. |
| returnLineNumber | int | Used to hold the line number of the Simple source code being processed. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHNEG |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **doGTJMP()** | | |
| **Name** | **Type** | **Description** |
| LHSAddress | int | Used to hold the address of the result of the expression on the left-hand side of the comparison operator. |
| RHSAddress | int | Used to hold the address of the result of the expression on the right-hand side of the comparison operator. |
| LHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the left-hand side of the comparison operator. |
| RHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the right-hand side of the comparison operator. |
| MAXLINELENGTH |  | Used to limit the length of the postfix expression arrays. Defined in SimpleCompiler.h |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| instruction1 | int | Used to create a LOAD instruction to load the first value of the comparison into the Accumulator. |
| instruction2 | int | Used to create a SUBTRACT instruction to second from the value stored in the Accumulator. |
| instruction3 | int | Used to create a BRANCHPOS instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is greater than zero. |
| returnLineNumber | int | Used to hold the line number of the Simple source code being processed. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. Defined in SMLOpCodes.h |
| BRANCHPOS |  | The value of an OPCode. Defined in SMLOpCodes.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **doLTEJMP()** | | |
| **Name** | **Type** | **Description** |
| LHSAddress | int | Used to hold the address of the result of the expression on the left-hand side of the comparison operator. |
| RHSAddress | int | Used to hold the address of the result of the expression on the right-hand side of the comparison operator. |
| LHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the left-hand side of the comparison operator. |
| RHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the right-hand side of the comparison operator. |
| MAXLINELENGTH |  | Used to limit the length of the postfix expression arrays. Defined in SimpleCompiler.h |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| instruction1 | int | Used to create a LOAD instruction to load the first value of the comparison into the Accumulator. |
| instruction2 | int | Used to create a SUBTRACT instruction to second from the value stored in the Accumulator. |
| instruction3 | int | Used to create a BRANCHNEG instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is negative. |
| Instruction4 | int | Used to create a BRANCHZERO instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is equal to zero. |
| returnLineNumber | int | Used to hold the line number of the Simple source code being processed. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHNEG |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHZERO |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **doGTEJMP()** | | |
| **Name** | **Type** | **Description** |
| LHSAddress | int | Used to hold the address of the result of the expression on the left-hand side of the comparison operator. |
| RHSAddress | int | Used to hold the address of the result of the expression on the right-hand side of the comparison operator. |
| LHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the left-hand side of the comparison operator. |
| RHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the right-hand side of the comparison operator. |
| MAXLINELENGTH |  | Used to limit the length of the postfix expression arrays. Defined in SimpleCompiler.h |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| instruction1 | int | Used to create a LOAD instruction to load the first value of the comparison into the Accumulator. |
| instruction2 | int | Used to create a SUBTRACT instruction to second from the value stored in the Accumulator. |
| instruction3 | int | Used to create a BRANCHPOS instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is greater than zero. |
| Instruction4 | int | Used to create a BRANCHZERO instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is equal to zero. |
| returnLineNumber | int | Used to hold the line number of the Simple source code being processed. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHPOS |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHZERO |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **doEQJMP()** | | |
| **Name** | **Type** | **Description** |
| LHSAddress | int | Used to hold the address of the result of the expression on the left-hand side of the comparison operator. |
| RHSAddress | int | Used to hold the address of the result of the expression on the right-hand side of the comparison operator. |
| LHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the left-hand side of the comparison operator. |
| RHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the right-hand side of the comparison operator. |
| MAXLINELENGTH |  | Used to limit the length of the postfix expression arrays. Defined in SimpleCompiler.h |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| instruction1 | int | Used to create a LOAD instruction to load the first value of the comparison into the Accumulator. |
| instruction2 | int | Used to create a SUBTRACT instruction to second from the value stored in the Accumulator. |
| Instruction3 | int | Used to create a BRANCHZERO instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is equal to zero. |
| returnLineNumber | int | Used to hold the line number of the Simple source code being processed. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHZERO |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |

|  |  |  |
| --- | --- | --- |
| **doNEJMP()** | | |
| **Name** | **Type** | **Description** |
| LHSAddress | int | Used to hold the address of the result of the expression on the left-hand side of the comparison operator. |
| RHSAddress | int | Used to hold the address of the result of the expression on the right-hand side of the comparison operator. |
| LHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the left-hand side of the comparison operator. |
| RHSPostfix | char[][] | Used to hold a postfix expression created from the expression on the right-hand side of the comparison operator. |
| MAXLINELENGTH |  | Used to limit the length of the postfix expression arrays. Defined in SimpleCompiler.h |
| returnAddress | int | Used to hold the address assigned to the return line number address within the SML Memory space where program execution will branch to. |
| instruction1 | int | Used to create a LOAD instruction to load the first value of the comparison into the Accumulator. |
| instruction2 | int | Used to create a SUBTRACT instruction to second from the value stored in the Accumulator. |
| instruction3 | int | Used to create a BRANCHNEG instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is negative. |
| Instruction4 | int | Used to create a BRANCHPOS instruction to branch the flow of execution to the returnAddress if the value in the Accumulator is greater than zero. |
| returnLineNumber | int | Used to hold the line number of the Simple source code being processed. |
| MEMORY | Int array | Used to store all of the instructions, variables and constants necessary to running an SML program. Defined in SimpleCompiler.h |
| INSTRUCTIONCOUNTER | int | Used to hold the address with the SML Memory space where the next instruction can be added. Defined in SimpleCompiler.h |
| LOAD |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| SUBTRACT |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHNEG |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| BRANCHPOS |  | The value of an OPCode. It is used to switch to a function creating the necessary instructions for the specified operation. Defined in SMLOpCodes.h |
| Flags | int [] | Used to hold the SML addresses of instructions which need their operands, eg BRANCH addresses, resolved during the second pass. Defined in SimpleCompiler.h |