**Report**

1. **INTRODUCTION**

The assembler is a program written in c language that converts the assembly code to machine language

It is written in c programming language

1. **Example**

The assembly code is passed as a test code to the assembler and then verified its complete output that is it

1. tokenized first
2. low level conversion achieved or not

Here is sample assembly code that is passed to assembler

**Assembly for**

.data

count: 60

array: .space 10

char: 0xfe

.code

ldi 0 count

ld 0 0

ldi 1 array

ldi 2 char

ld 2 2

lpp st 1 2

inc 1

dec 0

jz loop

jmp lpp

loop sub 1 2 3

lp1 jmp lp1

**The machine code of the above assembly program is listed below.**

000 1000 ldi 0 count

001 000f

002 2000 ld 0 0

003 1001 ldi 1 array

004 0010

005 1002 ldi 2 char

006 001b

007 2012 ld 2 2

008 3088 st 1 2

009 7609 inc 1

00a 7700 dec 0

00b 4001 jz loop

00c 5ffb jmp lpp

00d 71d1 sub 1 2 3

00e 5fff jmp lp1

00f 003c Variable count, with initial value 60, or 0x3c

010 0000 Ten empty locations of the variable array

011 0000

012 0000

013 0000

014 0000

015 0000

016 0000

017 0000

018 0000

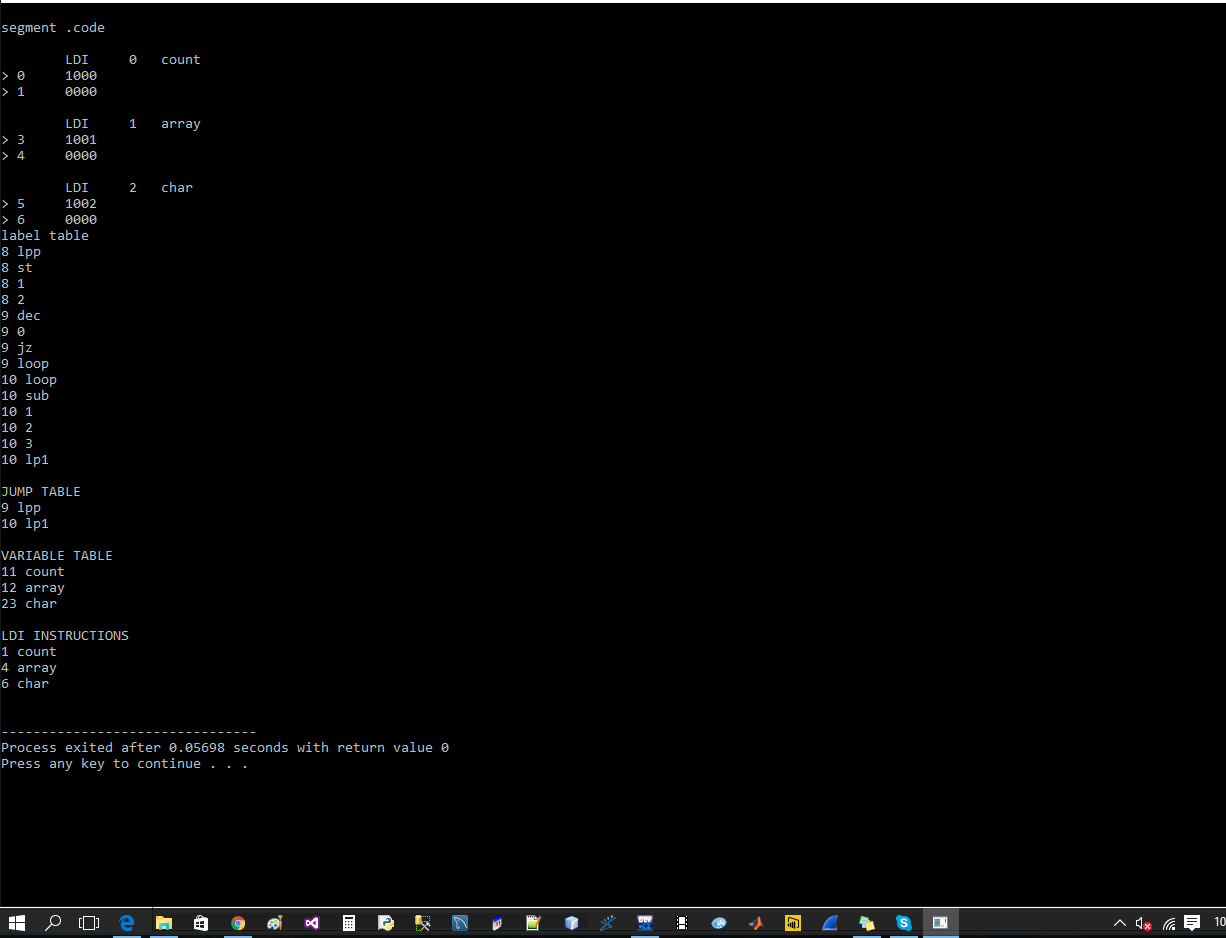
019 0000

01a 0000

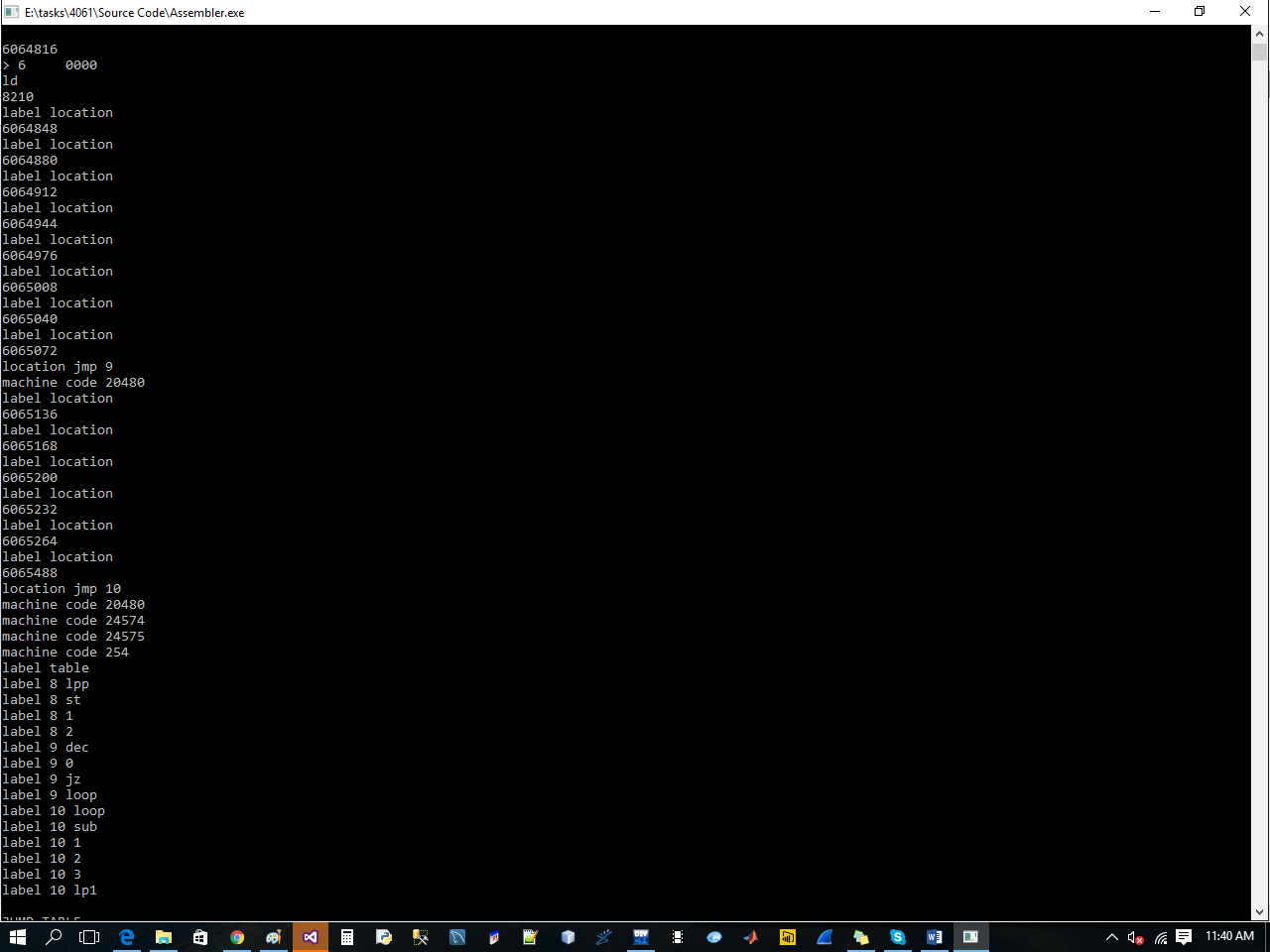
01b 00fe The variable char

1. **The Assembler Code**

First part is to get tokens and verifying the count here is the output of tokenized assembly code



The next part is related to get the machine code of its input token the part of the assembler code is modified to get the remaining output that is tokenized form to machine code here its output



That is representing each label , alu , add , pusp , pop , section , jump , variable and other fields like ldi in machine instruction as they executed they are printed

1. **How to use the assembler**

The part of code reads the assembly file is

fp = fopen("code.asm","r");

the assembly file contains the assembly code which instructions can be then converted to machine language

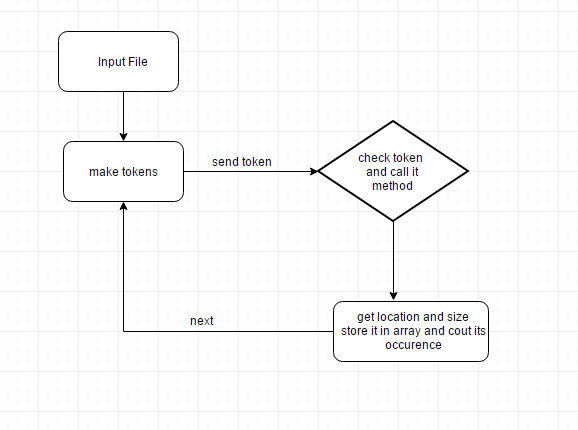
The program also have exe file and developed on windows platform using c language with provided assembly code

1. **Table for Assembly code token**

Below are tokens that assembler can process and show the conversion in output screen

|  |  |
| --- | --- |
| **ldi** | Ldi() method and lditable for location and label in c code |
| **Ld** | Ld() method |
| **jmp** | Jmp method and jumptable for location and label |
| **add** | Add method variabletable for location |
| **section** | Section method |
| **labelCall** | 14060304 current label location in execution |
| **not** | Logical operator also checked |
| **inc** |  |
| **and** | And logical operation also check |
| **pop** | Push and pop operation will be checked |
| **push** |  |
| **space** | Spaces ignored |
| **code** | .Code section also checked |
| **Ram** | Ram also checked |

1. **Flow chart**



1. **Summary**

The assembler converts the assembly language to machine language. The human readable form to machine code the output of the assembler that is developed by us show the count of fields and well as their encoding. The output images are attached that have been taken during development process of 2 steps tokenization then conversion to machine language.