

Experiment No.	04
Experiment Title	<b>Design and Implement Pass 1 of a Macro processor</b>  <b>Task: Write a program to read a given text file which can identify the macros in the program and display MNT, MDT and Argument List Array.</b>
Student Name	Vedant Ghag
Roll No.	23
Objectives:	<b>1 Students will be able to learn and identify macro definition And macro name in a given program.</b>  <b>2 Students will be able to understand the use of Macros in a program</b>
Theory /Algorithm:	<p>Procedure ExtractMacrosFromAssembly(assemblyFilePath):</p> <ul style="list-style-type: none"><li>Define a regular expression pattern for macro names</li><li>Define a regular expression pattern for macro definitions</li></ul> <p>Open the assembly file at assemblyFilePath for reading Read the entire content of the assembly file into a string</p> <p>Find all macro definitions in the content using the macro definition pattern For each macro definition found:     Extract the macro name and definition     Print the macro name and definition in a tabular format</p> <p>    Save the macro name to a file named "macro_[name]_name.txt"     Save the macro definition to a file named "macro_[name]_definition.txt"</p> <p>End For End Procedure</p> <p>Call ExtractMacrosFromAssembly with the path to the assembly file</p>

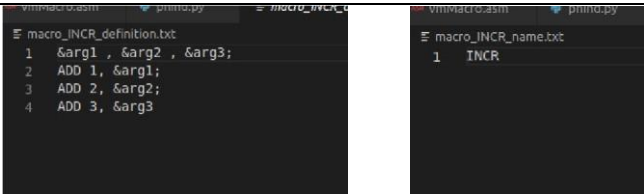
---

CSL601: System Programming and Compiler Construction Lab 2023-24

<b>Program Code:</b>	<pre>import re import os  # Define the regular expression patterns for macro names, definitions, and parameters</pre>
	<pre>macro_name_pattern = re.compile(r'MACRO\s+(\w+)') macro_definition_pattern = re.compile(r'MACRO\s+(\w+)\s+(\. +?);\s+MEND', re.DOTALL)  # Open the assembly file with open('vmMacro.asm', 'r') as file:    content     = file.read()  # Find all macros in the content macros = re.findall(macro_definition_pattern, content)  # Print and save macro names and definitions for i, (name, definition) in enumerate(macros, start=1):     print(f'Macro {i}:')    print(f'Name: {name}')     print(f'Definition: {definition}')  # Save the macro name and definition to separate files with open(f'macro_{name}_name.txt', 'w') as name_file:     name_file.write(name)      with open(f'macro_{name}_definition.txt', 'w') as definition_file: definition_file.write(definition)  # Print the table header print("\nMacro Name\tMacro Definition")  # Print the table rows for name, definition in macros:     print(f'{name}\t\t{definition}')</pre>

---

CSL601: System Programming and Compiler Construction Lab 2023-24

<b>Input to the Program:</b>	<pre>PG1 START MACRO INCR &amp;arg1 , &amp;arg2 , &amp;arg3; ADD 1, &amp;arg1; ADD 2, &amp;arg2; ADD 3, &amp;arg3; MEND MACRO DECR &amp;arg1 , &amp;arg2 , &amp;arg3; ADD 1, &amp;arg1; ADD 2, &amp;arg2; ADD 3, &amp;arg3; MEND INCR 4,5,6 ; END</pre>
<b>Output of the program:</b>	
<b>Outcome of the Experiment:</b>	To extract macro names, definitions, and input parameters from an assembly program using Python, you can use the re module for regular expressions. The script will read the assembly file, find macro definitions, and print and save the macro names and definitions in a tabular format. Each macro name and definition will be saved to separate files using the open function. The script will also handle the extraction of input parameters for each macro.
<b>References:</b>	Stack Overflow: <a href="https://stackoverflow.com/questions/14869400/pythonmacro-in-regular-expression">https://stackoverflow.com/questions/14869400/pythonmacro-in-regular-expression</a>