Assignment - Macro Preprocessor Ready Reference

An assembly language macro is a facility for extending the set of operations provided in an assembly

language. A programmer can define his own set of macros only once and can use them many times.

A macro definition consists of a name, a set of formal parameters and a body of code. When a macro name along with a set of actual parameters is used, it is replaced by body of macro and it is called macro expansion.

Macro processor is software that takes as input a program containing macro definitions and calls and generates an assembly language program which is free of macro definitions and where macro calls have been properly expanded. Macro processor has two main steps

* 1. Processing macro definitions
  2. Macro expansion

In the first step each macro definition is processed to extract the information and is stored in well-defined data structures. In macro expansion each macro call is expanded using appropriate information from the tables.

**Data Structure Design** – The design of macro processor requires several tables. The first preprocessing step uses tables such as macro name table, Keyword parameter default value table, macro definition table, parameter name table. Macro expansion step uses actual parameter name table, macro name table, Keyword parameter default value table and macro definition table. Since additions are to be done to all these tables we need pointers indicating last vacant position in the table.

|  |  |
| --- | --- |
| **Component** | **Description** |
| Macro name table (MNT) | It stores the name of the macro and other information such as no of positional parameters, keyword parameters etc.  It is used as a lookup table when a macro call is identified. It also contains  pointers to all other tables where relevant information is stored |
| Parameter name table (PNT) | It contains names of formal parameters including positional and keyword parameters |
| Keyword parameter default value table  (KPDT) | It contains keyword parameters and their default values |
| Macro definition Table  (MDT) | It contains the model statements of all macros. They are kept in partially  processed (IC) form so that expansion is easier |
| Actual parameter Table  (APT) | It contains actual parameters i.e. values that will replace formal  parameters during the expansion |

**Procedural Design** – The following table explains the input, algorithm and implementation hints for some of the procedures

|  |  |
| --- | --- |
| **Procedure** | **Description** |
| Macro Processor(main) | Input – source file as command line argument |
| Open the file and read line by line For each macro definition  Extract information from macro header statement and add to respective tables Store macro definitions statements in MDT till mend statement are reached by replacing parameters by positional markers.  For each macro call extract name and actual parameters  Expand after replacing macro definition statements corresponding to macro name by actual parameters |
| Extract | Extract macro name  Extract all positional parameters that start with & and add to PNT table  Extract all keywords that start with = and add keyword parameters and default values to KPDT table  Add name, parameter count and pointers to MNT table |
| Expand | Extract macro name from macro call Check if it is present in MNT table Get mdtptr and kpdptr from table  Prepare actual parameter table and add default values Extract and add actual parameters  Appropriately revert positions by parameters in statements in MDT table starting  from mdtptr till MEND |
| Replace | if parameter is present in macro body replace it by (P,n) where n is the parameter  position |
| Revert | Replace (P,n) in macro body by parameter name at nth position in actual  parameter name table |

Specifications Macro Pre-processor

Contents of Actual Parameter Table (APT) for first macro call expansion

|  |
| --- |
| Parameter Name |
| X |
| Y |

**FIRST expanded macro is**

|  |  |  |  |
| --- | --- | --- | --- |
| + | MOVER | AREG, | X |
| + | ADD | AREG, | Y |
| + | MOVEM | AREG, | X |

**Contents of Actual Parameter Table (APT) for second macro call expansion**

|  |
| --- |
| Parameter Name |
| X |
| Y |
| SUB |

**SECOND expanded macro is**

|  |  |  |  |
| --- | --- | --- | --- |
| + | MOVER | AREG, | X |
| + | SUB | AREG, | Y |
| + | MOVEM | AREG, | X |

**Output for Try it Program Handle the following set of errors:**

1. Invalid macro name in the macro call.
2. Insufficient number of actual parameters
3. Keyword parameters should not precede positional parameters in the actual parameter list.

Contents of Macro Definition Table (MDT)

|  |  |  |
| --- | --- | --- |
| MOVER | (P,3), | (P,1) |
| MOVEM | (P,3), | (P,2) |
| MEND |  |  |
| MOVER | (P,3) | (P,1) |
| (P,4) | (P,3) | (P,2) |
| MOVEM | (P,3) | (P,1) |
| MEND |  |  |

**Contents of Macro Definition Table (MNT)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Macro Name | #PP | #KP | MDTP | KPDTP |
| COPY | 2 | 1 | 0 | 0 |
| CHANGE | 2 | 2 | 3 | 1 |

**Contents of Keyword Parameter Default Value Table (KPDTAB)**

|  |  |
| --- | --- |
| Parameter Name | Value |
| REG | BREG |
| REG | AREG |
| OP | ADD |

**Contents of Parameter Name Table (PNTAB)**

|  |
| --- |
| Parameter Name |
| ONE |
| TWO |
| REG |
| FIRST |
| SECOND |
| REG |
| OP |

**Assignment No. 7 Macro Preprocessor Macro Expansion**

Practical Assignment

1. Consider the following Macro definition and its Data structures (stored as an array of structure) as follows:

MACRO

CALC &A,&B,&OP=ADD

MOVER AREG, &A

&OP AREG, &B

MOVEM AREG, &A MEND

Data Structures:

**Macro Definition Table** MOVER AREG, (P, 1) (P,3) AREG, (P,2) MOVEM AREG, (P,1) MEND

Parameter Name Table

|  |
| --- |
| Parameter Name |
| A |
| B |
| OP |

**Macro Name Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Macro Name | MDTPTR | KPDTPTR | No. Keyword Parameters | No. Positional Parameters |
| CALC | 0 | 0 | 1 | 2 |

**Keyword Parameter Default Value Table**

|  |  |
| --- | --- |
| Parameter Name | Default Value |
| OP | ADD |

Write a program to expand the above macro definition using given data structures for the following macro call.

* 1. CALC X,Y
  2. CALC X,Y,&OP=SUB

Do you Know it ?

1. What are the fields of MDT, KPDT, MNT, PNT, APT?
2. Explain and compare positional and keyword parameters. Explain how both can be used?
3. What is generated statement?
4. Compare macro with subroutine / function.
5. Explain nested macros.

|  |  |  |
| --- | --- | --- |
| **Assignment Evaluation** |  | |
| 0: Not Done | 1: Incomplete | 2: Late Complete |
| 3: Needs Improvement  **Viva –Voce (0 to 2)** | 4: Complete | 5: Well Done |
| **Signature of the Teacher** | **Signature of Student** | **Date** / / |
| Try it! |  |  |

1) Write a program to declare macro pre-processor data structures MNT, KPDTAB and PNTAB as static array of structure for the following macro definition:

MACRO

COPY &ONE, &TWO, &REG=BREG MOVER &REG, &ONE

MOVEM &REG, &TWO MEND

MACRO

CHANGE &FIRST, &SECOND, &REG=AREG, &OP=ADD MOVER &REG, &FIRST

&OP &REG, &SECOND MOVEM &REG, &FIRST MEND

Also write a program to expand the above macro definition using these data structures for the following macro call.

1. COPY A,B
2. CHANGE X,Y