Prita Hardasmalani
C14-2103058

THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

EXPERIMENT No. 7.

Aim: a) write a program to implement Pass 1 of MVlti-Pass Assembler

b) write a program to implement pass 2 of MVlti-pass Assembler.

Theory:

a] Pass1 of multipass assembler is responsible for scanning the source code, building a symbol table & assigning addresses to labels. The symboltable is crucial for resolving symbolic references & calculating the addresses of instructions and data items in subsequent passes.

1) Scan the source code:- Read the source code line by line
-Remove comment & whitespace from each line.

2) Identify labels: - check each line for the presence of labels.
- If a line contains a label, record it's name & current address

3] Generate symbol table:- create a symbol table to store label & their corresponding addresses.
-Assign addresses to labels based on their position in the source code. Handle cases of duplicate labels or invalid label names.

4] Increment location counter: Increment the location counter for each line encountered, regardless of whether it contains label or not. The location counter represents the address of the next instruction or data item.

5] Return symbol table: After scanning the entire source code, return the symbol table generated during pass 1. The symbol table will be used in subsequent passes for resolving symbolic references & generating to final machine code.

Example:

assembly language code ('source.code.asm')

```
START   LDA   VAL1     ; Load accumulator with value1.
        ADD   VAL2     ; Add value 2 to accumulator
        STA   RESULT   ; Store the result in memory
        HLT            ; Halt the program
VAL1    DAT   1        ; Data: value 1
VAL2    DAT   2        ; Data: value 2.
RESULT  DAT   0        ; Data: Result
```

pass 1 of the assembler using python.

```
def pass1_assembler (source_code):
    symbol_table = {}
    location_counter = 0
    with open (source_code, 'r') as file:
        for line in file:
            line = line.split (';') [0].strip()
            if not line: continue
            tokens = line.split()
            if ':' in taken [0]:
                label = token [0] [:-1]
                if label in symbol_table:
                    print ("Error: Duplicate label {}". format)
                return none
                symbol_table [label] = location_counter
            else:
                location_counter += 1
    return symbol_table.
```

6] Pass 2 of multi-pass assembler involves translating the assembly instruction into machine code using the symbol table generated in pass 1. Additionally, pass 2 performs error checking & generates the final machine code output.

1] Scan the source code - Read the source code line by line

2] Translate instruction: For each instruction encountered, translate it into machine code. Replace symbolic operand with their corresponding address from the symbol table generated in pass 1.

3] Generate machine code: concatenate the translated machine code instruction into a single output file or data structure. Ensure proper formatting & alignment of machine code instruction.

4] Error checking: Perform error checking during translation, such as detecting undefined symbols or invalid instruction.

5] Return machine code: After processing the entire source code, return the final machine code o/p.

Continuing the previous example,

pass 2 of the assembler using python.

```
def pass2_assembler(source_code, symbol_table):
        machine_code = []
        with open (source_code,'r') as file:
            for line in file:
                line = line.split (';')[0].strip()
                if not line:
                    continue.
        tokens = line.split()
        translated_instruction = ''
        for token in tokens:
            if token in symbol_table:
                translated_instruction +=
                            str(symbol_table[token]) + ' '
            else:
                translated_instruction += token + ' '
    machine_code.append(translated_instruction.strip())
return machine_code.
```

18/3/24

A+