

Aayush Shah

19BCE245

22 October 2022

Compiler Construction

Practical 6

Generate Three address code

• Code :

Prac6.1

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h" %}
%%
[0-9]+ {yylval.symbol = yytext[0]; return NUMBER;}
[a-zA-z]+ {yylval.symbol=yytext[0]; return LETTER;}
\n {return 0;}
. {return yytext[0];}
%%
yywrap(){
return 1;
}
```

Prac6.y

```
% {
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void convertToThreeAddressCode();char addToTable(char,
char, char);
int i = 0;
```

```

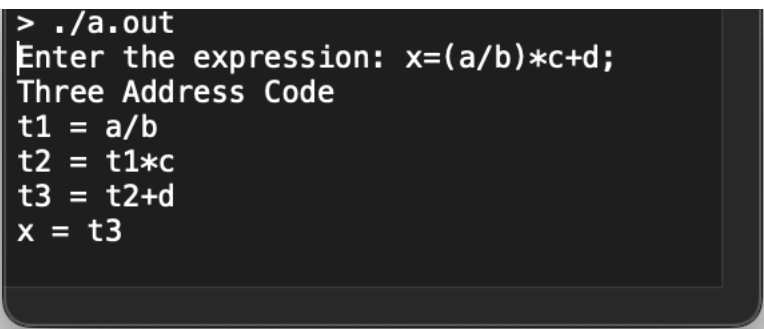
    char tmp = '1'; struct exp {
        char op1, op2, op;
    }; %
} %
union {
    char symbol;
} %
token < symbol > LETTER NUMBER %
    type < symbol > e %
    left '+'
    '-' %
    left '*'
    '/'
    '%' %
    %
stmt: LETTER '='
e ';' {
    addToTable($1, '=', $3);
} | e ';';
e: e '/'
e {
    $$ = addToTable($1, '/', $3);
} |
e '*'
e {
    $$ = addToTable($1, '*', $3);
} |
e '%'
e {
    $$ = addToTable($1, '%', $3);
} |
e '+'
e {
    $$ = addToTable($1, '+', $3);
} |
e '-'
e {
    $$ = addToTable($1, '-', $3);
} |
e '('
e ')' {
    $$ = (char) $2;
} |

```

```

NUMBER {
    $$ = $1;
} |
LETTER {
    $$ = $1;
}; %
%
yyerror(char * s) {
    printf("%s", s);
    exit(0);
}
struct exp code[20];
char addToTable(char op1, char op, char op2) {
    code[i].op1 = op1;
    code[i].op = op;
    code[i].op2 = op2;
    i++;
    return tmp++;
}
void convertToThreeAddressCode() {
    printf("\nThree Address Code\n\n");
    int cnt = 0;
    char tmp = '1';
    while (cnt < i) {
        if (code[cnt].op != '=')
            printf("t%c = ", tmp++);
        if (isalpha(code[cnt].op1))
            printf("%c ", code[cnt].op1);
        else if (code[cnt].op1 >= '1' && code[cnt].op1 <=
'9') printf("t%c ", code[cnt].op1);
        printf("%c ", code[cnt].op);
        if (isalpha(code[cnt].op2))
            printf("%c \n", code[cnt].op2);
        else if (code[cnt].op2 >= '1' && code[cnt].op2 <=
'9') printf("t%c \n", code[cnt].op2);
        cnt++;
    }
}
main() {
    printf("\nEnter the expression: ");
    yyparse();
    convertToThreeAddressCode();
}

```

output

```
> ./a.out
Enter the expression: x=(a/b)*c+d;
Three Address Code
t1 = a/b
t2 = t1*c
t3 = t2+d
x = t3
```