

21001715 컴파일러

Programming Assignment #5

Due: 2020년 6월 21일 23:59PM

Three Address Code 만들기

본 프로젝트에서는 PA1, 2 만들었던 문법들, 그리고 이를 이용해서 PA3에서 만든 Abstract Syntax Tree를 바탕으로 Three Address Code를 만든다.

1. Your Mission

C 언어로 작성된 소스코드가 있을 때, 이에 대한 3주소 코드를 작성하는 것이다. 이 때, 3주소 코드로 변환 가능한 C 언어의 문법은 PA4와 같다. 즉, PA3에서 함수의 선언 부분만 빼 문법들로 구성된 C 언어를 3주소 코드로 변환하는 것이 PA5의 목표이다.

2. 작성 방법

본 프로젝트에서 3주소 코드를 표현하기 위한 3주소 코드의 명령어는 다음과 같다.

(1) ***(a)** (단, a는 변수. 앞으로 이것을 포인터 변수라고 할 것임)

(2) **Var = a op b** (단, Var는 변수, a와 b는 변수 또는 상수, **op**는 +, -, *, /, %, >, >=, <, <=, ==, != 중 하나)

`fprintf(yyout, "\t%s\t= %s\t+ %s\n", ...);` 를 이용해서 출력할 것

(3) **Var = a** (단, Var는 변수, a는 변수 또는 상수. Var와 a는 포인터 변수도 가능한데, 이 경우는 둘 중 하나만 포인터 변수이어야 함.)

Var와 a가 둘 다 포인터 변수가 아니면 `fprintf(yyout, "\t%s\t= %s\t\n", ...);` 를 이용해서 출력

Var가 포인터 변수면 `fprintf(yyout, "\t*(%s)\t= %s\t\n", ...);` 를 이용해서 출력

a가 포인터 변수면 `fprintf(yyout, "\t*(%s)\t= *(%s)\t\n", ...);` 를 이용해서 출력

(4) **IFZ a Goto LB** (단, a는 변수 또는 상수, LB는 라벨)

`fprintf(yyout, "\tIFZ %s Goto %s\n", ...);` 를 이용해서 출력

(5) **Goto LB** (단, LB는 라벨)

`fprintf(yyout, "\tGoto %s\n", ...);` 를 이용해서 출력

(6) **LB:** (단, LB는 라벨)

`fprintf(yyout, "%s:\n", ...);` 를 이용해서 출력

(7) **BeginFunc**

`fprintf(yyout, "\tBeginFunc");` 를 이용해서 출력

(8) **EndFunc**

`fprintf(yyout, "\tEndFunc");` 를 이용해서 출력

(9) **PushParam a** (단, a는 변수 또는 상수)

`fprintf(yyout, "\tPushParam %s", ...);` 를 이용해서 출력

(10) **PopParam n** (단, n은 양의 정수)

`fprintf(yyout, "\tPopParam %d", ...);` 를 이용해서 출력

(11) **LCall** *f* (단, *f*는 함수 이름)

반환값이 있으면 `fprintf(yyout, "\t%s\t= LCall %s", ...)`; 이용해서 출력

반환값이 없으면 `fprintf(yyout, "\t%sLCall %s", ...)`; 이용해서 출력

(12) **Return** *a* (단, *a*는 변수 또는 상수이거나 없어도 됨)

반환값이 있으면 `fprintf(yyout, "\tReturn %s", ...)`; 이용해서 출력

반환값이 없으면 `fprintf(yyout, "\tReturn", ...)`; 이용해서 출력

3. 파일 설명

(1) `bar.y`

`main` 함수를 다음과 같이 수정한다:

```
extern FILE *yyin;
extern FILE *yyout;
ASTNode *root = 0;
stack = initStack();

yyin = fopen(argv[1], "r");
yyparse();
fclose(yyin);

root = pop(stack);
yyout = fopen(genOutputFN(argv[1]), "w");
buildTAC(root);
fclose(yyout);
return 0;
```

(2) `ast.h`

`void setName(ASTNode *n, char *nickName)`

- `node n`에 별명을 붙여주는 함수. 이 때, `n`이 `makeASTNodeID`, `makeASTNodeINT`, `makeASTNodeREAL` 중 어느 하나에 의해서 만들어진 `node`이면 이미 별명이 붙어 있으므로 수행할 필요 없다.

`char *getName(ASTNode *n)`

- `node n`의 별명을 반환하는 함수.

(3) `tac.h`

`LBSTACK`

- Label을 관리하기 위한 `stack` 타입

`char* mkTmp(void)`

- 임시 변수 (`_t`로 시작하는 것)를 생성해주는 함수

`char* genLabel(void)`

- Label (`L`과 정수로 구성되어 있음)을 생성해주는 함수

`void pushLabel(LBSTACK *lbStack, char *label)`

- Label `stack`에 `label`을 `push`하는 함수

`char *popLabel(LBSTACK *lbStack)`

- Label `stack`에서 `label` 하나를 `pop`하는 함수

`char *topLabel(LBSTACK *lbStack)`

- Label `stack`에서 맨 위에 있는 `label`을 반환하는 함수.

(4) tac.c

```
char* genTAC (ASTNode *node)
```

- 여러분이 주로 작업해야 하는 함수로, 재귀함수를 기반으로 함. 이해를 돕기 위하여 약간의 예제 코드를 넣었음.

4. 실행 예

컴파일 하면 c-- 라는 실행 파일이 만들어짐. 다음 코드 (파일명: s2.c) 가 있다고 하자:

```
int main(void) {
    int a[10], i, s;
    for(i=0; i<10; i++) {
        a[i] = 2*i + 1;
    }
    i = s = 0;
    while(i<10) {
        s += a[i++];
    }
    print(s);
    return 0;
}
```

./c-- s2.c 라고 입력하면 다음과 같은 내용의 s2.t라는 파일이 만들어진다:

```
main:
    BeginFunc
    i      = 0
    _t0     = i
L0:
    _t1     = i      < 10
    IFZ _t1 Goto L1
    _t2     = i      * 4
    _t3     = a      + _t2
    _t4     = 2      * i
    _t5     = _t4    + 1
    *( _t3) = _t5
    _t6     = *( _t3)
    _t7     = i
    i      = i      + 1
    Goto L0
L1:
    s      = 0
    _t8     = s
    i      = _t8
    _t9     = i
L2:
    _t10    = i      < 10
    IFZ _t10 Goto L3
    _t11    = i
    i      = i      + 1
    _t12    = _t11   * 4
    _t13    = a      + _t12
    _t14    = *( _t13)
    s      = s      + _t14
    _t15    = s
    Goto L2
L3:
    PushParam s
    _t16 = LCall print
    PopParam 4
    Return 0
    EndFunc
```

5. 제출

- 6월 21일 일요일 23:59까지. 시계는 제출용 서버의 시계를 따른다.
- 여러분의 계정에 PA5 디렉토리가 있고, 이 안에 `Makefile`, `ast.h`, `tac.h`, `tac.c`, `util.a`, `s0.c`, `s2.c` 파일들이 있을 것이다. `tac.c` 파일의 내용을 바꾸면 된다.
- 만일 위의 파일들 중 어느 하나를 지웠으면 담당 교수에게 문의할 것.
- 하루씩 Delay 될 때마다 점수의 20%를 감점한다. (예를 들어, 이를 Delay하고 8점을 받았으면 4.8점 획득)
- Copy는 해당 PA 0점. 본인 소스코드는 반드시 본인이 모두 타이핑 하여 만들어야 함. 오해를 방지하기 위하여 둘 이상이 같이 의논하여 째더라도 최대한 달라보이게 짤 것! (변수 바꾸기, 주석 더 넣기, 들여쓰기 바꾸기 등 만으로는 안 됨)