**2022-2 시스템 소프트웨어**

**텀프로젝트 보고서**

AI컴퓨터공학과 202111477 김지윤

* **선택한 기능 모듈**

: **Access symbol table**

: pass1의 모듈 중 하나로 symtab에 label, location과 data type을 저장한다.

* **모듈 설계**

1. **symtab 구조체 정의**

: label이름을 저장할 char형 배열 label, location의 값을 저장할 int형 변수 loc, 데이터 타입을 명시할 char형 배열 data\_type로 이루어진다. 200개를 선언한다.

1. **pass\_one 함수**

: **void pass\_one(FILE\* fp)**

: symtab의 전체적인 흐름을 주도한다. symtab을 만들기 위한 각종 함수를 호출할 함수이다.

1. fgets 함수를 통해 입력 파일인 f의 내용을 char형 배열인 buf에 한줄씩 읽어온다.
2. **get\_code** 함수를 호출, buf의 내용을 통해 char형 배열인 label, opcode, operanad를 설정
3. opcode의 내용이 START일 경우 strtoul 함수를 이용, 16진수로 변환 후 정수형 변수인 locctr과 startaddress에 저장한다.
4. opcode의 내용이 END와 comment가 아닐 경우
5. **search\_symbol** 함수를 호출, opcode의 내용이 symtab에 이미 있으면 에러
6. label에 내용이 있고 중복이 아닌 경우 label의 내용을 symtab의 label에 저장
7. LOCCTR의 내용을 symtab의 loc에 저장
8. opcode의 내용이 BYTE거나 RESB일 경우 data\_type에 “BYTE”를, WORD이거나 RESW일 경우 “WORD”를 저장
9. symtab에 string형태의 값을 저장할 때는 strcpy함수를 이용한다.
10. 사이즈 증가
11. **search\_op**함수 호출, opcode이면 **location\_counter** 함수 호출
12. 입력 파일의 내용이 끝날 때까지 반복
13. **get\_code 함수**

**: void get\_code(char\* buf)**

: pass\_one함수의 b단계에서 호출한 함수로, char형 배열로 선언한 label, opcode, operand 변수를 초기화한다.

1. buf의 내용을 받을 char형 포인터 변수 cp를 선언하고 opcode, label, operand를 공백으로 초기화한다.
2. strtok 함수를 이용하여 탭이나 다음줄으로 가기 전까지 buf의 내용을 끊는다. 예를 들어 buf의 내용이 “START 0” 이면 START까지를 의미한다. cp가 이를 가리키게 한다.
3. atoi 함수를 이용하여 cp의 내용을 정수형으로 변환한다. 문자열의 내용이 숫자면 정수형 숫자로 바꾸고 문자면 변환되지 않는다.
4. cp의 내용이 숫자면 반복문을 통해 buf에서 현재 cp 다음의 내용을 읽고 다시 cp가 가리키게 한다.
5. cp의 내용이 알파벳 이외의 기타 문자라면 break
6. cp의 내용이 알파벳일 때 search\_op 함수를 호출, opcode이면 cp의 내용을 opcode에 복사한다. RSUB 이외의 명령어에 대해 buf에서 cp다음의 내용을 cp에 저장하고 cp의 내용을 operand에 저장한다.
7. opcode가 아닌 경우 label이므로 cp의 내용을 label에 저장, 다음 내용들을 순서대로 opcode와 operand에 저장한다.
8. **search\_symbol 함수**

**: int search\_symbol(char\* cp)**

: pass\_one 함수의 d) 단계에서 호출한 함수로 symtab 배열에서 label과 같은 값이 있는지 찾는 함수이다.

1. char형 포인터 변수 dp를 선언, cp의 내용을 복사한다.
2. strcmp 함수를 이용, symtab에 저장된 label의 모든 내용과 비교하기 위해 for문으로 symtabsize만큼 반복하여 값이 같으면 TRUE를, 반복문이 끝나면 FALSE를 반환한다.
3. **search\_op 함수**

**: int search\_op(char \*cp)**

: pass\_one 함수의 e단계, get\_code 함수의 c단계에서 호출한 함수로 전달받은 값이 optab에 있는지 찾는 함수이다.

1. optab에 저장되어 있는 opcode의 수만큼(sizeof(optab)/sizeof(optab[n])) 반복, cp의 값과 같은 값이 있으면 TRUE를 반환, 반복문이 끝나면 FALSE를 반환한다.
2. **location\_counter 함수**

**: void location\_counter(char\* opcode)**

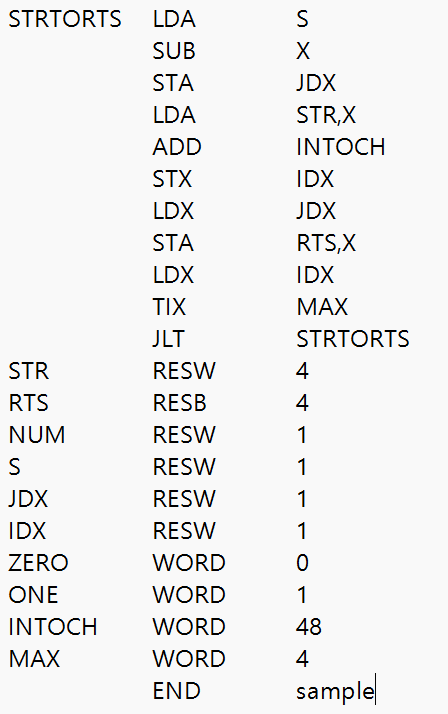
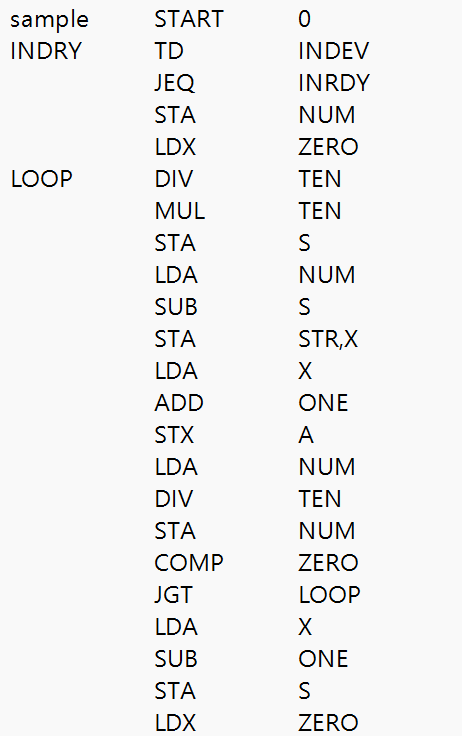
: location counter의 역할을 한다. locctr 변수의 값을 연산한다(변수는 바이트 단위라고 가정한다).

1. opcode의 값이 WORD이면 +3을 한다.
2. opcode의 값이 RESW이면 operand의 값을 정수형으로 변환 후 그 값만큼 3을 곱하고 더한다. ex) ARRAY RESW 30 🡪 locctr+= 3\* 30
3. opcode의 값이 BYTE이면 ‘C’일 때 ‘X’일 때로 나눈다.
4. ‘C’일 때, strlen함수를 이용, operand의 길이만큼 더한다.
5. ‘X’일 때, strlen함수를 이용한 후 /2를 한다.
6. ‘C’와 ‘X’ 를 빼야 하므로 -3을 한다.
7. opcode의 값이 지시어가 아닌 명령어일 경우 +3을 한다.

* 실행결과

: sample.txt 파일을 입력하여 symtab안에 잘 저장되었는지 확인한다.

입력🡪



실행 결과🡪

