

Ucode translator

2016112158 김희수



[날짜]

[회사 이름]

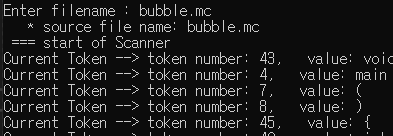
[회사 주소]

지난 과제에서 만든 AST파일을 이용해 ucode로 변환하는 translator를 만드는 과제이다. translator를 만들기 위해선 먼저 symbolTable이 구현되어야 한다. symbolTable에 대한 테이블을 초기화하는initSymbolTable(), symbolTable에서 원하는 엔트리를 찾기 위한 hash와 lookup, 그리고 엔트리 삽입을 위한 insert(), overflow가 일어났을 때 symbolTable을 dump하는 dumpSymbolTable(), block처리를 위한 set(), reset(), 마지막으로 symbolTable을 생성하는 genSym()이 있다.

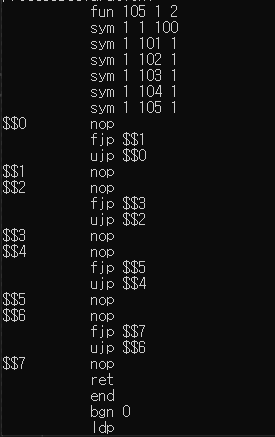
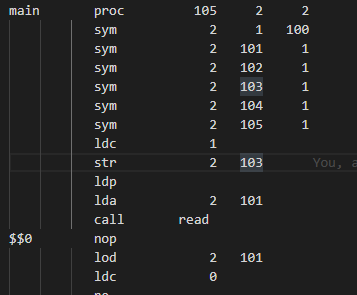
이제 Ucode를 프린트하는 emit함수를 살펴보자. sym을 찍어주는 emitSym, fjp,ujp,tjp등을 찍는 emitJump(), 함수를 찍어줄 때 사용하는 emitFunc(), add, sub 같은 operator를 찍는 emit0, opeartor와 operand하나를 찍어주는 emit1(), operator와 operand 둘을 찍어주는 emit2(), 점프를 위한 $$0, $$1등을 생성하는 genLabel()과 이를 찍어주는 emitLabel(). 마지막으로 값이 있는 리턴을 찍는 rv\_emit()이 있다

이 emit()함수들은 AST에 맞게 해당 ucode를 찍어주게 된다. 이제 AST를 처리하는 함수를 보자. AST는 leftson-right brother형태의 linkedList이므로 AST를 처리하는 process함수들은 전부 파라미터로 Node\* ptr을 갖는다. Ucode생성의 시작은 codeGen()을 통해 이루어진다. codeGen()에서 먼저 선언부를 processDeclaration()과 processFuncHeader()를 통해 처리한다. processDeclaration()에선 현재 노드의 typeSpecifier와 qualifier를 결정하고 그냥 단순 변수인지 어레이변수인지에 따라 그에 따른 process함수를 호출한다. 그 후, processFunction()을 통해 함수를 처리한다. processFunction()은 1. 형식매개변수 처리, 2. 함수몸체의 선언부 처리 3. function start code emit 4. 함수몸체의 { ~ } 처리. 이 과정에서 processStatement()를 이용한다. processStaement()에선 노드의 token.number에 따라 processOperator()를 사용해서 일반적인 문장을 처리할지 processCondtion()을 사용해서 if문과 while문을 처리할지 결정한다 5. 마지막으로 return 처리로 구성된다.

실험으론 bubble.mc 파일을 이용했다



생성된 Ucode는 다음과 같은데

정상적으로 출력되지 않았다. 오른쪽이 정상적으로 Ucode가 만들어진 bubble.uco이다. 정상적으로 출력되지 않는 이유는 AST가 잘못만들어진 거 같다. processStaement()에서 COMPOUND\_ST 케이스로 들어가면서 다시 processStaetment()를 호출하는데 이 두번째 호출에서 정상적이라면 EXP\_ST케이스로 들어가서 porcessOperator()를 호출해야한다. 하지만 어쨰서인지 WHILE\_ST케이스로 들어간다. 이는 AST의 문제라고 밖에 생각되지 않는다.