

시스템 프로그래밍 개인 과제

2020.10.06

과제 개요

-시스템 프로그램 교재의 2 장 figure 2.4 의 알고리즘을 응용하여 SIC/XE machine 의 assembler 를 구현

과제 기한

- 최종 보고서 및 결과물 제출 기한 : 2020 년 10 월 30 일 23:59 자정

문제

- 구현할 assembler 는 교재 Figure 2.4 에 예시된 2-pass algorithm 을 활용하여 구현.
- 구현된 assembler 는 기본적으로 SIC/XE machine 의 모든 instruction 과 directive (START, END, BYTE, WORD, RESB, RESW, BASE) statement 그리고 address mode, modification record 를 처리.
- 프로그램 실행은 다음과 같이 명령을 통해 input filename 과 output filename 을 받음.
- 구현한 어셈블러의 최종 파일명은 sicxe_assembler 이어야 함
- E.g) ./sicxe_assembler source1.asm object1.obj

구현 사항

- Assembler 파일 입출력 포맷 관련
 - assembler 는 다음과 같은 하나의 input file(source file)을 입력 받고 3 개의 output file (object file, assembly listing file, intermediate file)을 생성.
 - Source file: assembler 에게 입력되는 assembly source file (Figure 2.5 참조)
 - Object file: assembler 에 의해 생성되는 object program file (Figure 2.8 참조)
 - Assembly listing file: assembly listing file (Figure 2.6 및 교재 483page 참조)
 - Intermediate file: pass1 에서 pass2 로 전달되는 내부 data file
- Source file 의 format 는 아래와 같이 작성.
 - Label, Opcode, Operand, comment 는 탭 문자(\t) 또는 공백(space)으로 구분.

- 탭 문자(\t) 및 공백(space) 모두 처리 가능하여야함.
- 줄 바꿈 문자는 \n 사용
- 라인의 1 번째 byte 에 마침표(.)가 위치할 경우 라인 전체를 comment 로 간주

- Object file 은 교재의 object file format 을 준수.
- Intermediate file 은 assembler 의 pass 1 부분과 pass 2 부분간의 data 전달에 사용되므로 형식과 내용은 자유롭게 선택. (Pass 1 수행 후 생성)
 - Intermediate File 외 Pass 1 과 Pass 2 간의 정보 전달은 인정하지 않음
 - Hash Table 의 내용 출력 (Hash 구현 시)
 - Hash Table 사용 유무에 따른 소요시간 측정 결과 출력 (Hash 구현 시)
- Assembly listing file 은 입력되는 source file 의 처리 정보가 기록되어야 하며 format 은 교재 Figure 2.6 를 참고. Assembly listing file 에 반드시 기록해야 할 정보는 source file 의 각 라인에 대한 Loc, source code, object code, 발생한 error.
 - 교재 Figure 2.4 의 알고리즘에 명시된 3 가지 에러는 반드시 처리할 수 있어야 함. 그 이외의 에러는 처리는 자유롭게 구현하고 구현한 내용을 보고서에 명시.
 - duplicate symbol, invalid operation code, undefined symbol
- Hash 함수 관련 (선택 : 구현 시 가산점 부여)
 - OP code 들은 교재 Appendix A 의 순서대로 입력.
 - 외부 파일로부터의 입력 또는 헤더파일에서의 정의 등 Structure 정의 및 방법은 자유
 - OPTAB 및 SYMTAB 을 위한 Hash Table 형식을 구현
 - 외부인자를 통하여(옵션) Hash Table 사용 유무를 결정
 - 어떠한 Hash 함수를 사용하는가는 자유
 - Hash Table 사용시 OP code 는 Hashing 되어 Table 에 저장되고 탐색되어야 함
 - SYMTAB 에 대하여 Hash Table 사용 유무에 따른 시간 측정 함수를 구현.
 - 전체 알고리즘 수행에 대한 소요시간 측정
 - SYMTAB 에 대한 탐색 소요시간 측정 및 비교
 - 이러한 시간 측정을 수행할 수 있는 예제 코드 첨부파일(업로드 예정) 참조

. 기타 구현 사항

- Hash Table 사용 유무에 대한 소요시간 측정 결과 비교는 보고서의 시험평가 부분에서 기술.

- SIC/XE assembler 는 교재 2.1 과 2.2 절의 assembler 세부사항을 만족해야 함. 그밖에 교재 및 과제문서에서 제시하지 않은 기타 세부 형식은 자유롭게 정의하여 구현
- 사용언어는 C/C++(gcc)만 가능. 실행 환경은 Linux(Ubuntu 20.04)를 기준으로 함.

제출

- 소스 파일
- 컴파일 방법이 명시된 README 파일
- 결과 보고서
 - 아래 요구사항에 맞추어 PDF 파일로 제출
 - **형식 : 학번.pdf (e.g. 202004554.pdf)**
 - 요구사항
 - 실행 결과 스크린샷
 - 결과에 대한 설명
 - 구현사항 / 미구현사항 / 버그 / 개선점 등에 대한 설명
 - 미구현사항/버그의 경우 타당한 이유 제시
 - 고찰
- 위 제출물을 하나의 압축 파일로 압축하여(zip) 아주 BB 에 업로드하여 주시기 바랍니다.
 - **형식 : 학번_이름.zip (201204554_홍길동.zip)**
 - 제출물을 위 형식에 맞추지 아니한 경우 감점할 수 있음.
- 과제의 구현에 대한 질문 사항은 조교 이메일을 통해 해주시길 바랍니다.
- 각 보고서 작성시 불필요하거나 중복된 내용의 기술을 최대한 지양하여 주시고 여러분의 결과물을 설명하는데 필요한 내용을 중심으로 작성하여 주시기 바랍니다.
- 카피 관련
 - 다른 학생의 문서나 프로그램을 복제하여 제출할 경우 제출자와 제공자 모두 0 점 처리.
 - 이전 제출 목록 및 검색엔진으로 나오는 naver blog, tistory, github 등을 참고할 수 있으나 유사도가 지나치게 높을 경우 구현물에 대한 구두 테스트를 진행할 수 있음.
- 감점 사유
 - 프로그램의 구현을 완료하였더라도, 프로그램이 위 입력에서 요하지 아니하는 입력을 요구하는 경우 혹은 위 출력에서 요하지 아니하는 출력을 수행하는 경우
- 부분점수
 - 구현을 완료하지 못하였더라도 이를 보고서에 반드시 명시할것