과목명 : 시스템 프로그래밍

담당 교수명 : 김지환

<<Assignment 1>>

서강대학교 컴퓨터공학과

학번: 20171690

이름: 정유석

목차

1. 프로그램 개요
2. 프로그램 설명
   1. 프로그램 흐름도
3. 분할 c 파일
   1. 20171690.c
      1. 설명
      2. 모듈 정의
         1. 모듈 이름 : main()
            1. 기능
            2. 사용 변수
         2. 모듈 이름 : init()
            1. 기능
            2. 사용 변수
         3. 모듈 이름 : findCmd()
            1. 기능
            2. 사용 변수
         4. 모듈 이름 : removeSpace()
            1. 기능
            2. 사용 변수
   2. shell.c
      1. 설명
      2. 모듈 정의
         1. 모듈 이름 : help()
            1. 기능
            2. 사용 변수
         2. 모듈 이름 : directory()
            1. 기능
            2. 사용 변수
         3. 모듈 이름 : addHistory()
            1. 기능
            2. 사용 변수
         4. 모듈 이름 : history()
            1. 기능
            2. 사용 변수
         5. 모듈 이름 : quit()
            1. 기능
            2. 사용 변수
   3. memory.c
      1. 설명
      2. 모듈 정의
         1. 모듈 이름 : dump()
            1. 기능
            2. 사용 변수
         2. 모듈 이름 : edit()
            1. 기능
            2. 사용 변수
         3. 모듈 이름 : fill()
            1. 기능
            2. 사용 변수
         4. 모듈 이름 : reset()
            1. 기능
            2. 사용 변수
         5. 모듈 이름 : validAddr()
            1. 기능
            2. 사용 변수
         6. 모듈 이름 : validAddrRange()
            1. 기능
            2. 사용 변수
         7. 모듈 이름 : strToHex()
            1. 기능
            2. 사용 변수
         8. 모듈 이름 : isHex()
            1. 기능
            2. 사용 변수
   4. opcode.c
      1. 설명
      2. 모듈 정의
         1. 모듈 이름 : opcode()
            1. 기능
            2. 사용 변수
         2. 모듈 이름 : opcodeList()
            1. 기능
            2. 사용 변수
         3. 모듈 이름 : readOpTable()
            1. 기능
            2. 사용 변수
         4. 모듈 이름 : hashFunction()
            1. 기능
            2. 사용 변수
4. 구조체 정의
   1. HISTORY
   2. opNode
5. 전역 변수 정의
   1. HISTORY\* hisHead
   2. opNode\*\* opTable
   3. unsigned char\* MEMORY
   4. int END\_ADDR

1. 프로그램 개요

SIC/XE Machine을 구현하기 전, 어셈블러, 링크, 로더들을 실행할 Shell을 구현, object 코드가 적재되고 실행될 메모리 공간 할당, mnemonic과 그에 상응하는 opcode를 저장하는 OPCODE table 생성 들을 수행하는 프로그램이다. 프로그램은 입력 프롬프트 상태로 시작되며 Shell, memory, opcode 관련 명령어들을 수행한다.

2. 프로그램 설명

(그림)

2.1 프로그램 흐름도

3. 분할 c파일

위에서 기술한 것과 같이 프로그램이 수행하는 명령어는 Shell 관련, 메모리 관련, opcode 관련 명령어로 크게 세 가지로 나눌 수 있다. 이러한 기준으로 c 파일을 분할하였다. main 함수가 포함된 20171690.c 파일과 각각 관련 명령어 및 관련 기능을 수현한 shell.c, memory.c, opcode.c 파일이 존재한다.

3.1 20171690.c

3.1.1 설명

main함수가 포함된 c 파일이다. 프로그램 시작 후 전역변수를 초기화한다. 그 후 shell 형태의 명령 프롬프트에서 입력을 받아 유효한 입력인지 판단하고, parsing하여 적절한 기능을 수행한다.

3.1.2 모듈 정의

3.1.2.1 모듈 이름 : main()

3.1.2.1.1 기능

명령 프롬프트에서 입력을 받는다. 입력받은 문자열에서 command와 parameter들을 적절히 parsing하여 해당하는 명령어에 대한 동작을 수행(함수 호출)한다. 이 때 invalid한 입력에 대해 error message를 출력하고 입력을 다시 받는다.

한편 프로그램이 수행되며 input 변수에 dummy 값이 잔존하여 오류가 발생하는 상황을 방지하기 위해 매 순간 메모리를 새로 할당하고, 마지막에 해제한다.

3.1.1.1.2 사용 변수

char\* input – 명령 프롬프트에서 입력받은 문자열. 이 때 입력받는 문자열의 크기는 20171690.h 파일에 MACRO로 100으로 할당되었다.

char\* input\_formed – 입력받은 Input에서 앞, 뒤 white space를 제외한 문자열이다.

char\* cmd – input\_formed에서 strtok을 사용하여 command를 가리키는 포인터이다.

char \*params, \*start, \*end, \*addr, \*val – cmd와 같이 input\_formed에서 각각 parameters, start 주소, end 주소, 주소, 값을 가리키는 포인터이다. 이 때 각각 변수는 주어진 명령어에 상응하는, 필요한 parameter에 대한 값만 가리키며 잘못된 입력의 경우 원하는 값을 저장하지 못할 수 있다.

int before, after – 해당 변수는 “dump” 명령어에서만 사용된다. comma(,)에 의한 잘못된 입력이 들어왔을 경우 comma를 기준으로 문자열을 token으로 잘라 자르기 전의 문자열 길이(before)와 자른 후의 문자열 길이(after)를 비교한다. 둘의 길이가 다른 경우는 “dump start, ”와 같이 잘못된 입력을 의미하므로 에러 메시지를 출력한다.

3.1.2.2 모듈 이름 : init()

3.1.2.2.1 기능

프로그램을 처음 실행시켰을 때 전역변수를 초기화하고 필요한 메모리를 할당한다. opcode에 대한 hash table도 이 때 생성한다.

3.1.2.2.2 사용 변수

없음

3.1.2.3 모듈 이름 : findCmd(char\* cmd)

3.1.2.3.1 기능

main 함수에서 입력받은 input을 토큰 단위(해당 프로그램에서는 공백)로 자른다. 이 때 input에서 command에 해당하는 문자열을 parameter로 입력받아 원하는 command에 대한 코드를 반환한다. 상응하는 command가 없는 경우 main에서 에러 메시지를 출력한다.

3.1.2.3.2 사용 변수

char\* cmd – input에서 parsing한 명령어의 시작 주소를 가리킨다.

3.1.2.4 모듈 이름 : removeSpace(char\* input)

3.1.2.4.1 기능

parameter로 전달받은 문자열에 대해 시작과 끝부분에 있는 white space(‘ ‘, ‘\t’, ‘\n’)를 제거한다. 이 때 문자열의 시작 주소를 white space가 아닌 character가 처음 나오는 위치로 지정하고, 끝부분에는 white space가 아닌 가장 마지막 character 다음 위치에 null 문자(‘\0’)를 저장하는 방식이다. return 형은 char\*이며 새로 지정한 문자열의 시작 주소를 반환한다.

문자열의 시작 주소가 null인 경우, 바로 시작주소(null)을 반환한다.

3.1.2.4.2 사용 변수

char\* input – 앞과 뒷부분의 white space를 삭제할 문자열의 시작 주소이다.

3.2 shell.c

3.2.1 설명

3.2.2 모듈 정의

3.2.2.1 모듈 이름 : help()

3.2.2.1.1 기능

3.2.2.1.2 사용 변수

3.2.2.2 모듈 이름 : directory()

3.2.2.2.1 기능

3.2.2.2.2 사용 변수

3.2.2.3 모듈 이름 : addHistory()

3.2.2.3.1 기능

3.2.2.3.2 사용 변수

3.2.2.4 모듈 이름 : history()

3.2.2.4.1 기능

3.2.2.4.2 사용 변수

3.2.2.5 모듈 이름 : quit()

3.2.2.5.1 기능

3.2.2.5.2 사용 변수

3.3 memory.c

3.3.1 설명

3.3.2 모듈 정의

3.3.2.1 모듈 이름 : dump()

3.3.2.1.1 기능

3.3.2.1.2 사용 변수

3.3.2.2 모듈 이름 : edit()

3.3.2.2.1 기능

3.3.2.2.2 사용 변수

3.3.2.3 모듈 이름 : fill()

3.3.2.3.1 기능

3.3.2.3.2 사용 변수

3.3.2.4 모듈 이름 : reset()

3.3.2.4.1 기능

3.3.2.4.2 사용 변수

3.3.2.5 모듈 이름 : validAddr()

3.3.2.5.1 기능

3.3.2.5.2 사용 변수

3.3.2.6 모듈 이름 : validAddrRange()

3.3.2.6.1 기능

3.3.2.6.2 사용 변수

3.3.2.7 모듈 이름 : strToHex()

3.3.2.7.1 기능

3.3.2.7.2 사용 변수

3.3.2.8 모듈 이름 : isHex()

3.3.2.8.1 기능

3.3.2.8.2 사용 변수

3.4 opcode.c

3.4.1 설명

3.4.2 모듈 설명

3.4.2.1 모듈 이름 : opcode()

3.4.2.1.1 기능

3.4.2.1.2 사용 변수

3.4.2.2 모듈 이름 : opcodeList()

3.4.2.2.1 기능

3.4.2.2.2 사용 변수

3.4.2.3 모듈 이름 : readOpTable()

3.4.2.3.1 기능

3.4.2.3.2 사용 변수

3.4.2.4 모듈 이름 : hashFunction()

3.4.2.4.1 기능

3.4.2.4.2 사용 변수

4. 구조체 정의

4.1 HISTORY

4.2 opNode

5. 전역 변수 정의

5.1 HISTORY\* hisHead

5.2 opNode\*\* opTable

5.3 unsigned char\* MEMORY

5.4 END\_ADDR