구현과제 (80점)

아래에 나열된 24개의 32비트 RISC-V 기본 명령어들로 작성된 입력프로그램(*.s)이 주어지는 경우, 각 명령어를 정해진 규칙대로 32비트 기계어코드로 변환하고 (*.o 파일 생성), 시작 명령어 메모리 주소를 1,000번지로 가정할 때 실행되는 동안의 프로그램 카운터(PC) 값의 변화를 기록한 파일을(*.trace) 생성하는 프로그램을 구현하시오.

▸ 구현 조건

- 구현은 C 언어로만 제한하고, 실행 테스트는 Linux 환경에서 GCC로 테스트함
- 사용하는 명령어는 다음의 23개 RISC-V 기본 명령어 + 1개의 추가 명령어로 (EXIT) 제한함
    - ADD, SUB, ADDI, AND, OR, XOR, ANDI, ORI, XORI, SLL, SRL, SRA, SLLI, SRLI, SRAI, LW, SW, BEQ, BNE, BGE, BLT, JAL, JALR
    - EXIT: 코드 실행을 종료하는 명령어, 명령어 Format은 0xFFFFFFFF 로 가정함
- 범용 register는 x0~x31으로 번호로만 사용함. register 이름은 사용하지 않음
- x1, x2, x3, x4, x5, x6 register의 초기값은 각각 1, 2, 3, 4, 5, 6으로 가정하고, 나머지 register들의 초기값은 0으로 가정함
- 명령어의 대소문자 구분은 하지 않음

▸ 입력

- 프로그램을 실행하면 사용자는 바로 입력 파일을 받을 수 있도록 구현함
- 입력 파일이 존재하지 않는 경우에는 "Input file does not exist!!" 메시지를 출력하고 다시 파일명을 입력받도록 구현함
- 입력 파일에 기록된 Assembly 코드는 한 줄에 하나의 명령어가 기록되고, 공백 Line은 무시함. 입력프로그램의 Line 수는 최소 1줄 이상이며 최대 Line 수는 제한이 없음
- "terminate"가 입력된 경우에는 프로그램 수행을 종료함

▸ 출력

- 입력 파일에 문법에 맞는 Assembly 코드가 작성된 경우에는 파일명.o, 파일명.trace 파일을 생성하고 새로운 파일 입력을 대기함
- 입력 파일에 문법에 맞지 않는 Assembly 코드가 하나라도 존재하는 경우에는 "Syntax Error!!" 메시지를 출력하고 새로운 파일 입력을 대기함. 문법 오류가 발생한 경우에는 파일명.o, 파일명.trace 파일을 생성하지 않음

▸ 제출 요구사항

- 구현 및 테스트를 완료한 소스 코드는(C 파일) 하나의 파일로 제출해야 함 (압축 파일 제출 X)
- 보고서는 별도의 PDF 파일로 제출해야 함
- 소스 코드 GCC 컴파일 시 Option이 필요한 경우에는 반드시 보고서 첫째 줄에 Compile Option을 기재해야 함
- 보고서에는 본인이 구현한 코드를 개략적으로 설명해야 하고, 실행 결과 스냅샷은 반드시 포함하여야 함. 보고서에 소스 코드 전체를 포함할 필요는 없음
- 제출 요구사항 미준수 시에는 10% 감점 처리함

▸ 실행 예

>> Enter Input File Name: test1.s

test1.s

```
ADD x7, x1, x2
SUB x8, x3, x4
AND x9, x5, x6
OR x10, x7, x8
XOR x11, x9, x10
ANDI x12, x11, 11
ORI x13, x12, 15
XORI x14, x13, 12
SLL x15, x14, x2
SRL x16, x15, x3
SRA x17, x16, x4
SLLI x18, x17, 2
SRLI x19, x18, 1
SRAI x20, x19, 1
LW x21, 0(x1)
ADD x22, x1, x2
SUB x23, x2, x1
AND x24, x22, x23
OR x25, x22, x23
SW x21, 8(x2)
```

test1.o

```
00000000001000001000001110110011
01000000010000011000010000110011
00000000011000101111010010110011
```

00000000100000111110010100110011
00000000101001001100010110110011
00000000101101011111011000010011
00000000111101100110011010010011
00000000110001101100011100010011
00000000001001110001011110110011
00000000001101111101100000110011
01000000010010000101100010110011
00000000001010001001100100010011
00000000000110010101100110010011
01000000000110011101101000010011
00000000000000000101010101000011
00000000001000001000101100110011
01000000000100010000101110110011
00000001011110110111110000110011
00000001011110110110110010110011
00000001010100010010010000100011

test1.trace

1000
1004
1008
1012
1016
1020
1024
1028
1032
1036
1040
1044
1048
1052
1056
1060
1064
1068
1072
1076

>> Enter Input File Name: test2.s

test2.s

```
ADD x7, x1, x0
SUB x8, x7, x1
ADDI x8, x0, 9
ORI x9, x8, 45

LOOP:
ADD x7, x7, x8
SUB x9, x9, x8
BGE x9, x8, loop

ADDI x9, x0, 5
ADDI x7, x0, 7
BGE x7, x9, LAB

ADD x10, x1, x2
LAB:
SUB x11, x3, x4
```

test2.o

```
00000000000000001000001110110011
01000000000100111000010000110011
00000000100100000000010000010011
00000010110101000110010010010011
00000000100000111000001110110011
01000000100001001000010010110011
11111110100001001101110011100011
00000000010100000000010010010011
00000000011110000000000001110010011
00000000100100111101010001100011
00000000001000001000010100110011
01000000010000011000010110110011
```

test2.trace

```
1000
1004
1008
1012
1016
```

1020
1024
1016
1020
1024
1016
1020
1024
1016
1020
1024
1016
1020
1024
1028
1032
1036
1044

>> Enter Input File Name: test3.s

test3.s

```
ADD x7, x1, x2
SLLI x8, x7, 2
SRLI x9, x8, 1

JAL x1, func1
ADD x12, x1, x3
SUB x13, x4, x2
AND x14, x5, x6

LW x19, 4(x1)
ADD x20, x19, x2
SLLI x21, x20, 1
SRLI x22, x21, 2
SW x22, 4(x2)
EXIT

func1:
```

ADD x15, x1, x2
SUB x16, x3, x4
ADD x17, x5, x6
OR x18, x1, x2
JALR x0, 0(x1)

test3.o

00000000001000001000001110110011
00000000001000111001010000010011
00000000000101000101010010010011
00000010100000000000000000011101111
00000000001100001000011000110011
01000000001000100000011010110011
00000000011000101111011100110011
00000000010000001010100110000011
00000000001010011000101000110011
00000000000011010000110101010010011
00000000001010101101101100010011
00000001011000010010001000100011
11111111111111111111111111111111
00000000001000001000011110110011
01000000010000011000100000110011
00000000011000101000100010110011
00000000001000001110100100110011
00000000000000000100000001100111

test3.trace

1000
1004
1008
1012
1052
1056
1060
1064
1068
1016
1020
1024

1028
1032
1036
1040
1044
1048