

MIPS Programming Assignment

4190.308-001 Computer Architecture (2012-fall)

Update: 2012-11-27

1. Introduction

1.1. 목표

본 과제는 컴퓨터구조 과목을 수강하는 학생들이 MIPS simulator인 SPIM을 이용하여 MIPS assembly programming을 경험해 보는 것을 목적으로 합니다.

1.2. 과제 주제

이번 과제는 Segment Tree를 이용한 point query algorithm을 MIPS assembly language로 programming하는 것입니다.

1.3. Due date

Due date는 2012년 11월 26일(월요일) 11:59 PM입니다. 1일 delay될 때마다 5%씩 감점되며 제출 기한 일주일(12월 3일) 후에는 과제를 받지 않습니다.

1.4. Development tool

SPIM에는 크게 SPIM, XSPIM, PCSPIM, QTSPIM이 있습니다. 이번 과제에서는 **QTSPIM**을 사용할 예정입니다. 조교 또한 QTSPIM으로 채점할 예정입니다. 부득이하게 SPIM이나 XSPIM, PCSPIM을 사용하실 분들은 QTSPIM에서도 검증해 보시고 보고서에 꼭 명시해 주시기 바랍니다.

2. 과제 설명

이번 과제는 text file (input.txt)에 있는 data를 입력으로 받아서 segment tree를 이용하여 point query를하여 결과를 console 창에 출력하는 것입니다. Assembly programming은 course homepage에 있는 C source code를 기반으로 작성하십시오. 그리고 과제는 아래와 같이 진행하시면 됩니다.

2.1 Main (10점)

- 프로그램이 처음 실행되고, 필요한 변수들을 할당합니다. Initialize함수를 호출, segment들을 입력 받고 insertSegment함수를 호출하여 입력 받은 segment들을 segment tree안에 넣습니다. 그 다음 query point q 를 console창으로 입력 받아 query함수를 호출하여 이 결과를 출력해주는 역할입니다.

```
Main
{
    // 1. initialize()

    // 2. file input & insertSegment()

    // 3. input query point

    // 4. query()

    // 5. print result
}
```

2.2 Preprocessing

2.2.1 Initialize

- 전체 range (RANGE_MIN, RANGE_MAX)를 가지고 segment tree를 만들어주는 함수입니다. 범위는 C코드에 주어진 대로 1~20으로 하시면 됩니다.
- tree구조가 올바르게 생성되었음을 확인할 수 있게 treeArray와 각 entry의 minRange, maxRange를 console창에 출력해주시기 바랍니다.

2.2.2 Insert segment

- 입력 받은 segment들을 segment tree에 넣어주는 함수입니다.
- File input을 하지 않을 경우 input data는 input.txt에 있는 data를 사용합니다.
 - 총 25개 input: (1,7), (2,6), (3,9), (4,15), (5,10), (6, 8), (7,13), (8,19), (10, 17), (11, 6), (12, 20), (13, 1), (14, 13), (15, 17), (16, 6), (17, 8), (18, 10), (19, 15), (20, 17), (1, 3), (3, 5), (4, 7), (8, 9), (16, 20)

2.3 Query

- 입력 받은 query point를 가지고 segment tree를 query하는 함수입니다.
- Query가 올바르게 되었음을 확인할 수 있게 queryArray를 console창에 출력해주시기 바랍니다.
 - queryArray 출력 시 초기화되지 않은 쓰레기 값을 출력하지 않기 위해 main 초반에 queryArray의 entry를 모두 '0'으로 초기화 해주시기 바랍니다.

2.4 FILE I/O

- 2.2과 2.3에서는 입/출력에 대해서는 고려하지 않습니다.
- 만약 file input 기능을 구현하지 못할 것 같으면 그냥 console로 입력 받거나 코드상에 input을 입력해주어도 됩니다. 이 경우 File I/O에 대한 점수는 없습니다.
- 2.4는 input.txt file을 입력으로 받아서 preprocessing을 한 후 point에 대한 query를 한 결과를 console창에 출력하는 것입니다.
- 주의 사항:
 - file 입출력은 ascii code로 이뤄집니다. 따라서 file input시 ascii code 형태로 입력 받은 data를 decimal로 변경해야 합니다. (file output을 구현할 경우 decimal을 ascii code로 변경한 후 file로 출력해야 합니다.)
 - 파일 입출력은 QtSpim.exe가 위치한 폴더에서 실행됩니다. Vista 이상에서 UAC를 사용하는 경우 파일 입출력이

C:\사용자\사용자ID\AppData\Local\VirtualStore\Program Files
(x86)\QtSpim"에서 실행될 수 있습니다. 이 점 유의해주시기 바랍니다.

2.5. 보고서 작성 (20점)

지금까지 실습한 내용을 보고서로 작성하십시오. 보고서의 양식이나 분량에는 제한이 없으나, 아래 내용은 반드시 포함되어 있어야 합니다.

- Segment tree에 대한 설명
 - 제공한 C code에 대한 설명이 아니라 인터넷, 책을 찾아보고 segment tree에 대한 내용을 설명 - field에서 이용되는 경우의 time complexity
 - ◆ 주어진 code의 time complexity는 분석하지 않아도 됩니다.
 - Segment tree로 query하는 과정 설명 및 initialize(), insertSegment(), query() 함수의 동작 설명 (pseudo code를 제공하면 좋습니다.)
- Program의 전반적인 설명 + 실행화면
 - Procedure call 시 calling convention: caller와 callee의 역할에 대해서 자세한 설명
- File I/O에 대한 설명
 - System call
 - Ascii code → decimal, decimal → ascii code로의 변환 과정
- 어려웠던 점 / 해결방법 / 느낀 점

3. 평가

평가는 100점을 만점으로 합니다. 그리고 due date 기준으로 1일 delay될 때마다 5%씩 점수를 깎겠습니다. 제출 기한 일주일 후에는 과제를 받지 않겠습니다. 평가 기준은 아래 table을 참조하시기 바랍니다.

Category	Score	Description
Main	10	<ul style="list-style-type: none">● Source code 정상 동작: 5● 주석 달기 (comment): 5
Preprocess	20	<ul style="list-style-type: none">● Source code 정상 동작: 15● 주석 달기 (comment): 5
Query	30	<ul style="list-style-type: none">● Source code 정상 동작: 25● 주석 달기 (comment): 5
File I/O	20	<ul style="list-style-type: none">● File input: 20
보고서	20	<ul style="list-style-type: none">● Segment tree 설명: 5● Program 설명 + 실행화면: 5● File I/O 설명: 5

4. 과제 제출

4.1. 제출 방법

- Source code와 보고서를 “**학번_이름.zip**”으로 압축하여 메일로 보내주시기 바랍니다.
- **메일주소:** ywchoi@archi.snu.ac.kr
- **파일:** 학번_이름.zip
- **MIPS Source code, 보고서** 포함

교과목 홈페이지에 Tutorial 링크가 있습니다. Programming을 하다가 궁금한 내용이 있으면 Tutorial 페이지나 example들을 참고하시고, 교과목 게시판이나 조교에게 메일을 보내주시면 답변 드리도록 하겠습니다.

5. Reference

- Wikipedia
- Section 10.3 in de Berg, Cheong, van Kreveld, Overmars; Computational Geometry: Algorithms and Applications, 3rd edition, 2008