

# **System Programming Project 2**

## 1. 개발 목표

- 해당 프로젝트에서 구현할 내용을 간략히 서술.
- (주식 서버를 만드는 전체적인 개요에 대해서 작성하면 됨.)
- 두 가지 방식으로 Concurrent한 주식 서버를 구현한다.
- 첫 번째 방식은 Event – driven approach이다. 이 방식은 하나의 프로세스가 모든 요청을 관리하는 방식이다. 두 번째는 thread - based server로서 thread pool을 활용하여 각각의 client에 하나의 working thread를 대응시키는 방식이다.
- 주식은 빠른 탐색을 위하여 binary search tree로 구현하였다. 서버를 구동하면 우선 stock.txt 파일을 읽어 트리를 구성한다. Client로부터 주식에 대한 요청(show, buy, sell)이 들어오면 트리를 순회하며 해당 주식을 처리하도록 하였다. Connection이 모두 종료되는 경우에는 stock.txt 파일에 업데이트된 주식 정보를 반영한다.

## 2. 개발 범위 및 내용

### A. 개발 범위

- 아래 항목을 구현했을 때의 결과를 간략히 서술
  - 1. Task 1: Event-driven Approach
    - Client의 요청에 따라 서버의 주식 정보가 업데이트 되고 결과가 stock.txt에 잘 반영이 되었다.
  - 2. Task 2: Thread-based Approach
    - Client의 요청에 따라 서버의 주식 정보가 업데이트 되고 결과가 stock.txt에 잘 반영이 되었다.
  - 3. Task 3: Performance Evaluation
    - Client의 요청이 서버에서 잘 반영되는 것을 확인한 후에 Event-driven approach와 Thread-based approach의 실제 성능을 테스트하여 I/O multiplexing과 thread synchronization이 어떤 경우에 더 효율적인지 확인하였다.

## B. 개발 내용

- 아래 항목의 내용만 서술
- (기타 내용은 서술하지 않아도 됨. 코드 복사 붙여 넣기 금지)
- Task1 (Event-driven Approach with select())
  - ✓ Multi-client 요청에 따른 I/O Multiplexing 설명
    - ◆ I/O Multiplexing은 여러 I/O 이벤트들을 처리하기 위해 사용된다. 이벤트는 connect 요청을 받는 것, fd가 데이터를 수신하는 것, client에 write가 가능한 것 등을 포괄한다. I/O Multiplexing은 select 함수를 통해 fd bit vector를 탐색하면서 이벤트가 발생한 fd를 검사한다. 이번 프로젝트에서 fd bit vector를 read\_set이라 하고 이벤트가 발생한 fd는 ready 상태가 되었다고 명시한다. Select 함수는 ready 상태가 된 fd의 개수를 반환한다. 그리고 각각의 fd에 대하여 I/O request를 처리한다.
  - ✓ epoll과의 차이점 서술
    - ◆ 위에서 설명한 select 함수는 read\_set에 대해 for loop를 돌며 이벤트 발생 여부를 확인한다. maxfd 변수를 통해 탐색할 범위를 줄이기는 하지만 select 함수가 호출될 때마다 for 문을 돌기 때문에 비효율적이다. epoll 함수는 이를 해결하기 위해 리눅스 상에서 운영체제가 직접 fd 저장소를 관리한다. 즉 epoll 함수는 이벤트가 발생한 fd들을 한데 모아 반환할 수 있고 그에 따라 select 함수의 항상 for문으로 인한 비효율성을 개선한다.
- Task2 (Thread-based Approach with pthread)
  - ✓ Master Thread의 Connection 관리
    - ◆ sbuf buffer를 통해 connection을 관리한다. Master thread는 sbuf를 초기화하고 while loop에 진입한다. Accept 함수를 통해 connected fd를 리턴받고 해당 connected fd를 sbuf에 삽입한다. 그리고 worker thread가 sbuf에서 connfd를 받아와 client의 요청을 처리한다.
  - ✓ Worker Thread Pool 관리하는 부분에 대해 서술
    - ◆ Thread 기반 서버를 thread pool 없이도 구현할 수 있다. 하지만 client로

부터 연결 요청이 올때마다 thread를 생성해야 하는 overhead가 발생하게 된다. 이를 해결하기 위해 미리 thread들을 생성한 후 thread pool에 보관한다. 그리고 sbuf buffer에서 삭제 가능한 connfd가 확인되면 thread pool에서 대기하던 thread가 해당 connfd의 요청을 처리한 후 다시 thread pool에서 대기한다.

#### - Task3 (Performance Evaluation)

- ✓ 얻고자 하는 metric 정의, 그렇게 정한 이유, 측정 방법 서술
  - ◆ 동시 처리율을 "초당 처리하는 request 개수"로 정의하였다. Concurrent server는 동시에 몇 개의 요청을 처리할 수 있는지가 관건이기 때문이다. 동시처리율을  $R$ , client 수를  $CNUM$ , multichannel 파일을 실행하여 측정한 결과를  $T(ms)$ 라 하면  $R = \frac{CNUM * ORDER\_PER\_CLIENT}{T * 10^6}$  이라 정의하였다.
  - ◆ Multichannel 파일을 이용하여 ORDER\_PER\_CLIENT, STOCK\_NUM, BUY\_SELL\_MAX 값을 고정시킨 후 client 개수에 변화를 주며 시간을 측정하였다. 또한 thread based server에서 thread 개수에 따른 변화를 측정하기 위해 ORDER\_PER\_CLIENT, STOCK\_NUM, BUY\_SELL\_MAX 값을 고정시킨 후 NTHREADS 값을 변화시키며 시간을 측정하였다. 정확도를 위하여 두 번씩 측정한 후 평균값으로 계산하였다.
- ✓ Configuration 변화에 따른 예상 결과 서술
  - ◆ SHOW, BUY, SELL 랜덤 요청 시 → Thread 방식이 event 방식보다 빠를 것이라 예상하였다. Event 방식은 하나의 프로세스가 모든 요청을 처리하는 반면 Thread 방식은 여러 thread들이 요청을 분담하여 처리하기 때문이다.
  - ◆ SHOW만 요청 시 → Random 요청을 보내는 경우보다 thread의 동시처리율이 증가하여 event 방식과 thread 방식 간의 동시처리율 차이가 작아질 것이라 예상하였다. 왜냐하면 show요청을 보낼 경우 thread의 경우는 자료를 읽기만 하면 되기 때문이다. Semaphore를 readers – writers problem을 기반으로 구현했기 때문에 reading에는 제한이 없기 때문이다.
  - ◆ BUY, SELL만 요청 시 → Random 요청보다 thread 방식의 동시처리율이 감소하여 event 방식과 thread 방식의 동시처리율 차이가 커질 것이라 예상했다. 왜냐하면 readers – writers problem 방식으로 semaphore를 구

현했기 때문에 자료에 write (buy, sell)를 수행하려면 더 많은 semaphore locking이 필요하기 때문이다.

- ◆ Thread based server에서 NTHREADS를 변화시킨 경우 → Thread의 개수가 많을수록 동시처리율이 그에 비례하여 높아질 것으로 예상하였다. 더 많은 thread들이 요청들을 분담할 수 있기 때문이다.

### C. 개발 방법

- B.의 개발 내용을 구현하기 위해 어느 소스코드에 어떤 요소를 추가 또는 수정할 것인지 설명. (함수, 구조체 등의 구현이나 수정을 서술)
- TASK 1
  - ✓ 기본 구성은 교재의 echoservers.c를 참고하였다.
    - ◆ Connfd를 저장하는 Struct pool 구조체를 선언하고 pool 변수를 초기화 하는 init\_pool 함수를 구현하였다. Connfd를 pool에 삽입하는 add\_client 함수 또한 구현하였다.
  - ✓ 추가 구현
    - ◆ 우선 트리를 구성할 수 있도록 addNodeToTree 함수와 createNode 함수를 구현하였다. Stock.txt 파일을 한 줄씩 읽을 때마다 createNode 함수를 호출한다. createNode 함수는 해당 주식 정보에 맞는 node를 반환한다. 이후 호출한 addNodeToTree 함수는 해당 node를 전달 받고 binary search tree에서 알맞은 위치를 검색하여 노드를 삽입한다.

```
TreeNode* createNode(int id, int amount, int price) {  
  
    TreeNode* node = (TreeNode*)malloc(sizeof(TreeNode));  
    node->stockItem.id = id;  
    node->stockItem.amount = amount;  
    node->stockItem.price = price;  
    node->left = node->right = NULL;  
  
    return node;  
}
```

```

void addNodeToTree(TreeNode* node) {

    if (root == NULL) {
        root = node;
        return;
    }
    TreeNode* temp = root;
    int newId = node->stockItem.id;

    while (true) {
        if (newId == temp->stockItem.id) {
            printf("stock id: %d - already exists. \n", newId);
            return;
        }
        if (newId < temp->stockItem.id) {
            if (temp->left == NULL) {
                temp->left = node;
                break;
            } else {
                temp = temp->left;
            }
        } else {
            if (temp->right == NULL) {
                temp->right = node;
                break;
            } else {
                temp = temp->right;
            }
        }
    }
    return;
}

```

- ◆ check\_clients 함수를 변형하여 트리를 조작할 수 있도록 하였다. check\_clients 함수가 buffer에서 읽은 요청에 따라 show, buy, sell을 실행 할 수 있도록 하였다. 아래는 check\_clients 함수에 추가한 코드이다.

```

/* If the descriptor is ready, echo a text line from it */
if ((connfd > 0) && (FD_ISSET(connfd, &p->ready_set))) {

    p->nready--;
    if ((n = Rio_readlineb(&rio, buf, MAXLINE)) != 0) {
        byte_cnt += n;

        strcpy(buf_copy, buf);
        buf_copy[strlen(buf_copy) - 1] = '\0';
        strcpy(cmd_experiment, buf_copy);
        //printf("Server received %d (%d total) bytes on fd: %d\n", n, byte_cnt, connfd);

        char *argv[10] = {0};
        int argc = parseline(buf_copy, argv);

        if (!strcmp(argv[0], "show")) {

            char newBuf[MAXLINE];
            memset(newBuf, '\0', sizeof(newBuf));
            createResultString(root, newBuf);
            Rio_writen(connfd, newBuf, MAXLINE);
        }
        else if (argc == 3) {
            int action_id = atoi(argv[1]);
            int action_amount = atoi(argv[2]);
            bool flag = false;

            if (!strcmp(argv[0], "sell")) {
                flag = true;
            }
            searchAndUpdate(action_id, action_amount, flag, connfd, cmd_experiment);
        }
    }
    else { /* EOF detected, remove descriptor from pool */
}

```

- ◆ searchAndUpdate 함수를 구현하였다. 해당 함수는 tree를 순회하며 buy, sell 요청을 처리한다.

```

void searchAndUpdate(int targetId, int amount, bool action, const int connfd, char* cmd) {
    char buf[MAXLINE];
    TreeNode* node = root;
    bool updated = false;

    while (node) {
        if (targetId == node->stockItem.id) {
            if (action || node->stockItem.amount >= amount) { // sell or buy stock
                if (action) {
                    node->stockItem.amount += amount;
                    sprintf(buf, "[sell] success\n");
                } else {
                    node->stockItem.amount -= amount;
                    sprintf(buf, "[buy] success\n");
                }
                updated = true;
            }
            break;
        } else if (targetId < node->stockItem.id) {
            node = node->left;
        } else {
            node = node->right;
        }
    }

    if (!updated) {
        sprintf(buf, "Not enough left stock\n");
    }
    Rio_writen(connfd, buf, MAXLINE);
    return;
}

```

- ◆ createResultString 함수를 구현하였다. 해당 함수는 show 요청을 받았을 경우 호출되며 tree를 순회하여 show에 대한 결과로 출력할 문자열을 생성한다. 또한 stock.txt 업데이트에 사용할 문자열 또한 생성하는데 사용된다.

```

void createResultString(TreeNode* node, char* newBuf) {
    if (!node) return;

    char tmp[10];

    createResultString(node->left, newBuf);

    sprintf(tmp, "%d", node->stockItem.id);
    strcat(newBuf, tmp);
    strcat(newBuf, " ");
    sprintf(tmp, "%d", node->stockItem.amount);
    strcat(newBuf, tmp);
    strcat(newBuf, " ");
    sprintf(tmp, "%d", node->stockItem.price);
    strcat(newBuf, tmp);
    strcat(newBuf, "\n");

    createResultString(node->right, newBuf);
}

```

## - TASK 2

- ✓ 기본 구성은 교재를 참고하여 작성하였다.

- ◆ Connection과 thread들을 관리하는 sbuf\_init, sbuf\_deinit, sbuf\_insert, sbuf\_remove, void \*thread 함수는 교재를 참고하였다.
- ✓ 추가 구현
  - ◆ 트리를 생성하는 createNode 함수와 addNodeToTree 함수는 Task1과 동일하다.
  - ◆ createResultString과 searchAndUpdate 함수도 동일하지만 Thread synchronization을 위하여 semaphore와 관련된 코드를 추가하였다. Readers-writers problem을 참고하여 트리의 정보를 읽는 부분은 reader 함수로, 트리를 수정하는 부분을 writer 함수라 가정하고 코드를 작성하였다. 아래는 searchAndUpdate 함수에서 semaphore를 추가한 부분과 semaphore를 추가한 createResultstring 함수이다.

```

while (node) {

    if (targetId == node->stockItem.id) {
        if (action || node->stockItem.amount >= amount) { // sell or buy stock

            P(&node->stockItem.w);
            if (action) {
                node->stockItem.amount += amount;
                sprintf(buf, "[sell] success\n");
            }
            else {
                node->stockItem.amount -= amount;
                sprintf(buf, "[buy] success\n");
            }
            V(&node->stockItem.w);
            updated = true;
            fprintf(stdout, "%s success \n", cmd);
            fflush(stdout);
        }
        break;
    }
}

```

```

void createResultString(TreeNode* node, char* newBuf) {

    if (!node) return;

    createResultString(node->left, newBuf);

    P(&(node->stockItem.mutex));
    (node->stockItem.readcnt)++;
    if (node->stockItem.readcnt == 1) /* First in */
        P(&(node->stockItem.w));
    V(&(node->stockItem.mutex));

    char tmp[10];
    sprintf(tmp, "%d", node->stockItem.id);
    strcat(newBuf, tmp);
    strcat(newBuf, " ");
    sprintf(tmp, "%d", node->stockItem.amount);
    strcat(newBuf, tmp);
    strcat(newBuf, " ");
    sprintf(tmp, "%d", node->stockItem.price);
    strcat(newBuf, tmp);
    strcat(newBuf, "\n");

    P(&(node->stockItem.mutex));
    (node->stockItem.readcnt)--;
    if (node->stockItem.readcnt == 0) /* Last out */
        V(&(node->stockItem.w));
    V(&(node->stockItem.mutex));

    createResultString(node->right, newBuf);
}

```

- ◆ echo 함수에 show, buy, sell 명령을 처리하는 코드를 추가하였다. Task1의 check\_clients 함수에서 처리한 방식과 동일하다. 다만 명령 처리 부분을 semaphore로 보호해주었다.

```

while((n = Rio_readlineb(&rio, buf, MAXLINE)) > 0) {
    P(&mutex);
    strcpy(buf_copy, buf);
    buf_copy[strlen(buf_copy) - 1] = '\0';
    strcpy(request, buf_copy);

    printf("server received %d bytes\n", n);

    buf[strlen(buf) - 1] = '\0';
    char *argv[10] = {0};
    int argc = parseline(buf, argv);

    if (!strcmp(argv[0], "show")) {

        char newBuf[MAXLINE];
        createResultString(root, newBuf);
        newBuf[strlen(newBuf)] = '\0';
        Rio_writen(connfd, newBuf, MAXLINE);
    }
    else if (argc == 3) {

        int action_id = atoi(argv[1]);
        int action_amount = atoi(argv[2]);
        bool flag = false;

        if (!strcmp(argv[0], "sell")) {
            flag = true;
        }
        searchAndUpdate(action_id, action_amount, flag, connfd, request);
    }
    V(&mutex);
}

```

### 3. 구현 결과

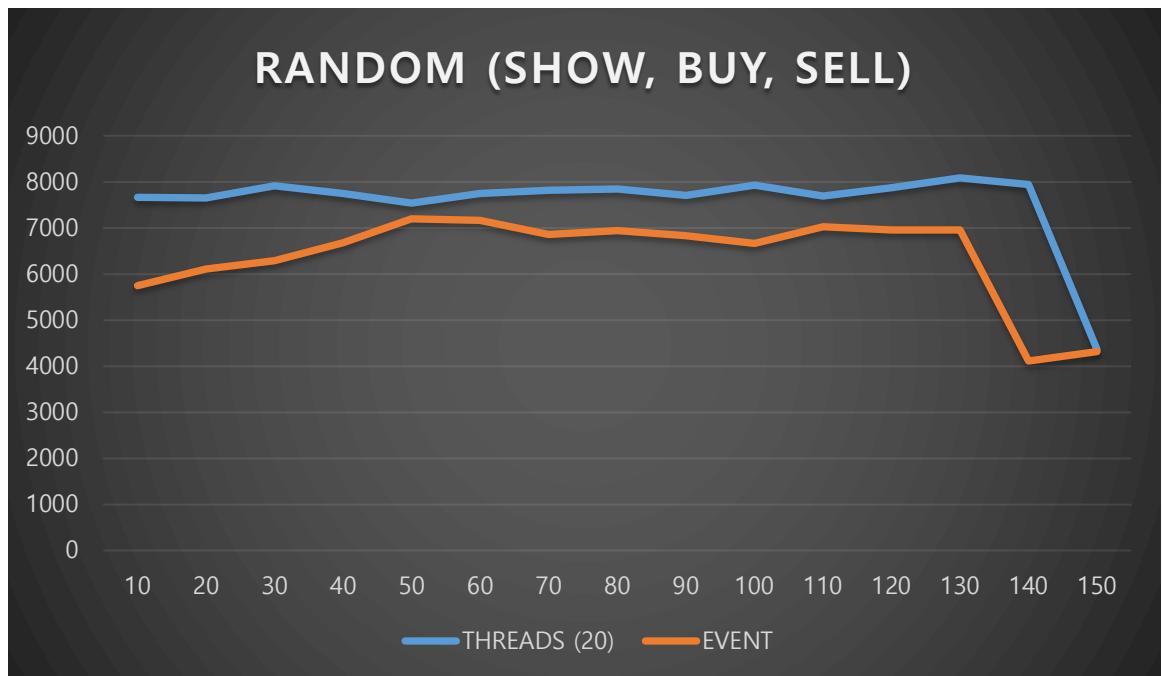
- 2번의 구현 결과를 간략하게 작성
  - ✓ Task1과 Task2 모두 client 요청 결과가 stock.txt에 잘 반영되는 것을 확인하였다.
- 미처 구현하지 못한 부분에 대해선 디자인에 대한 내용도 추가

### 4. 성능 평가 결과 (Task 3)

- 강의자료 슬라이드의 내용 참고하여 작성 (측정 시점, 출력 결과 값 캡처 포함)
- 기본 설정
  - ✓ ORDER\_PER\_CLIENT=30
  - ✓ STOCK\_NUM=5
  - ✓ BUY\_SELL\_MAX=10
  - ✓ Server=cspro, client=cspro8

- **RANDOM**

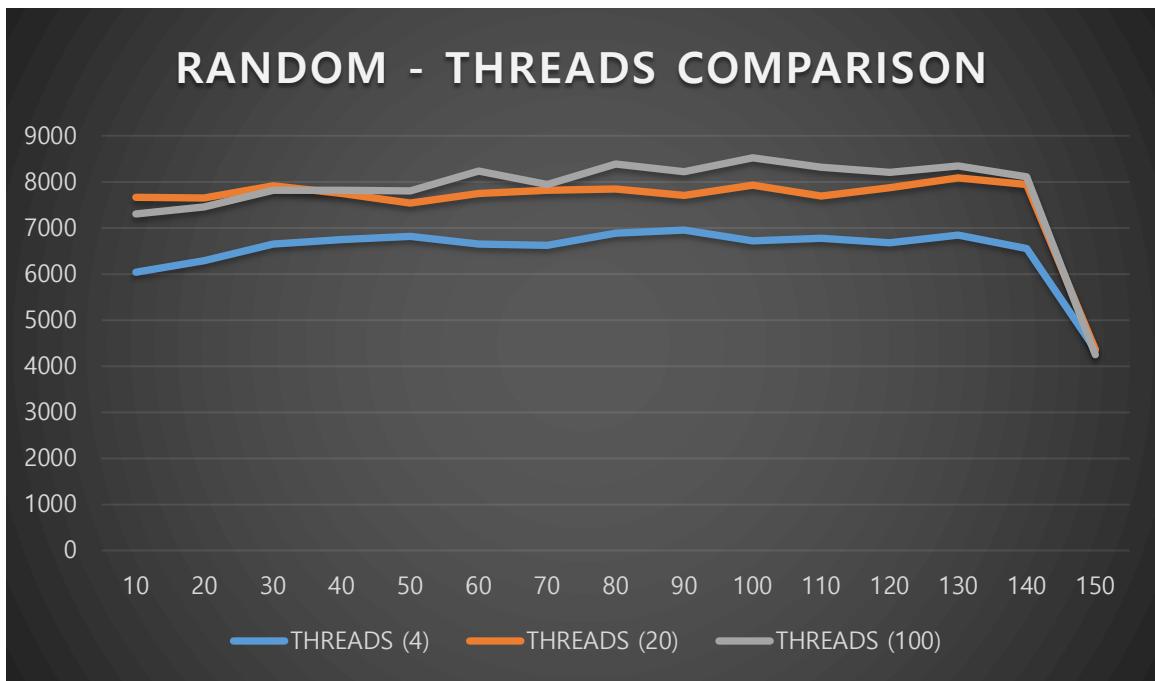
- ✓ Thread 방식이 event 방식보다 빠를 것이라 예상하였다. Event 방식은 하나의 프로세스가 모든 요청을 처리하는 반면 Thread 방식은 여러 thread들이 요청을 분담하여 처리하기 때문이다.



- ✓ 위는 실제 테스트 결과이고 예상대로 thread 방식이 동시처리율이 더 높은 것으로 나타났다. 두 방식 모두 client 수가 140 ~ 150 이상이 되는 지점에서 처리율이 급격히 감소했다. Thread 방식의 동시 처리율이 급감한 것은 synchronization overhead 때문이라 생각한다.

#### - RANDOM – THREAD #4 vs THREAD #20 vs THREAD #100

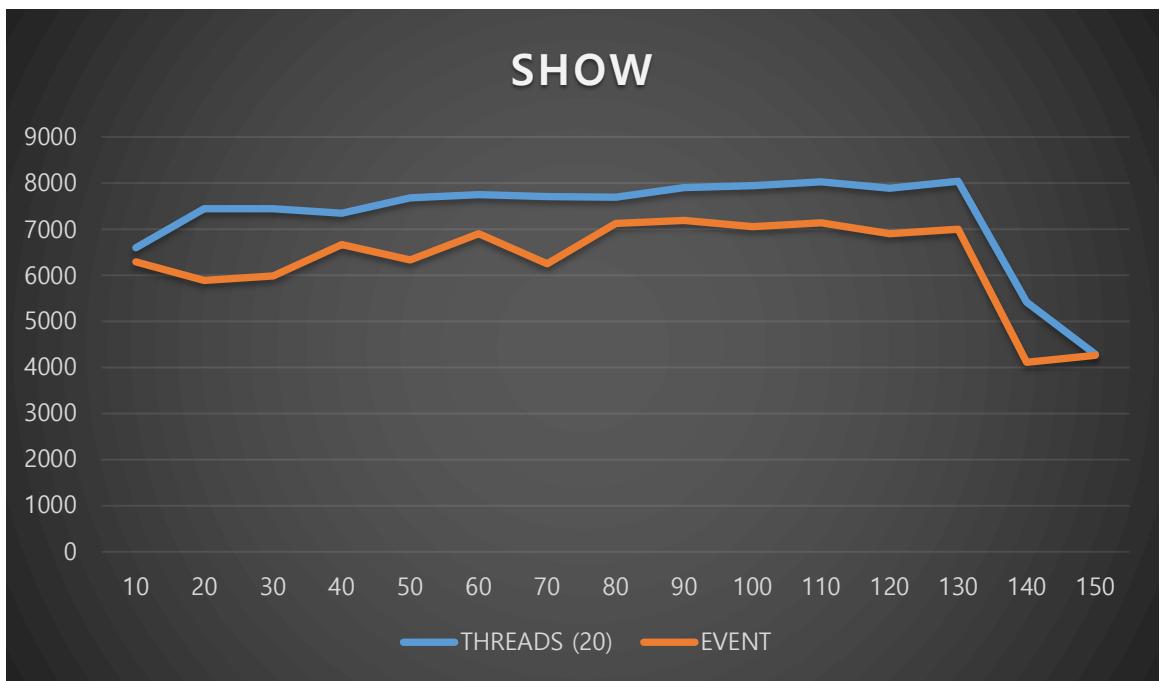
- ✓ 같은 조건하에서 thread의 개수만 변화시키며 실험을 진행하였다. Thread의 개수가 많을수록 동시처리율이 그에 비례하여 높아질 것으로 예상하였다. 더 많은 thread들이 요청들을 분담할 수 있기 때문이다.



- ✓ 위는 thread 개수에 따른 동시처리율 측정 결과이고 예상과 맞게 thread 개수가 많을수록 동시처리율이 높아지는 것으로 측정되었다. 하지만 thread 개수에 비례하여 동시처리율이 상승하진 않았다. 위 그래프를 보면 thread#20과 thread#100의 차이보다 thread#20과 thread#4의 차이가 더 크다. 이는 thread 개수의 증가에 따라 synchronization overhead도 그만큼 커지기 때문이라 생각한다.

#### - SHOW only

- ✓ Client가 show 요청만 보내는 경우를 실험하였다. 이는 Random 요청을 보내는 경우보다 thread의 동시처리율이 증가하여 event 방식과 thread 방식 간의 동시처리율 차이가 작아질 것이라 예상하였다. 왜냐하면 show요청을 보낼 경우 thread의 경우는 자료를 읽기만 하면 되기 때문이다. Semaphore를 readers – writers problem을 기반으로 구현했기 때문에 reading에는 제한이 없기 때문이다.



- ✓ 위는 show 요청만 보낸 경우의 실제 측정 결과이다. Random 요청의 경우와 유의미한 차이를 보이지는 않았으나 전 구간에서 동시처리율의 차이가 약간 더 일정한 것을 확인할 수 있었다.

- **BUY, SELL only**

- ✓ Client가 buy, sell 요청만 보내는 경우를 실험하였다. Random 요청보다 thread 방식의 동시처리율이 감소하여 event 방식과 thread 방식의 동시처리율 차이가 커질 것이라 예상했다. 왜냐하면 readers – writers problem 방식으로 semaphore를 구현했기 때문에 자료에 write (buy, sell)를 수행하려면 더 많은 semaphore locking이 필요하기 때문이다.



- ✓ 위는 client가 buy, sell 요청만 보냈을 경우를 실제 측정한 결과이다. Random 한 경우와 비교했을 때 큰 차이를 보이지 않았다. Stock 개수가 5개밖에 되지 않기 때문이라 생각한다.
- ✓ 다만 Random, Show only 경우와 다르게 client 개수가 증가할수록 event 방식과 thread 방식의 동시처리율 차이가 점점 작아지는 양상을 보였다. 특히 Show only 경우와 thread 방식을 비교해보면 Show only의 경우에는 thread 방식의 동시처리율이 client 수가 증가함에 따라 약간씩 상승하는 반면 BUY, SELL 의 경우는 그렇지 않을 수 있다. 이는 client 개수가 증가할수록 그만큼 synchronization overhead도 커지기 때문이라 생각한다. 따라서 thread 방식과 event 방식의 동시처리율이 client가 증가함에 따라 그 차이가 줄어든다고 생각한다.

## - 실험 캡쳐

```
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 10
[SHOW] elapsed time: 66768 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 20
[SHOW] elapsed time: 112087 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 30
[SHOW] elapsed time: 145047 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 40
[SHOW] elapsed time: 181779 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 50
[SHOW] elapsed time: 262281 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 60
[SHOW] elapsed time: 282765 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 70
[SHOW] elapsed time: 293858 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 80
[SHOW] elapsed time: 337309 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 90
[SHOW] elapsed time: 384174 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 100
[SHOW] elapsed time: 397289 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 110
[SHOW] elapsed time: 473632 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 120
[SHOW] elapsed time: 514532 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 130
[SHOW] elapsed time: 552377 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 140
[SHOW] elapsed time: 592401 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 150
[SHOW] elapsed time: 1055327 microseconds

cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 10
[BUY_SELL] elapsed time: 45806 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 20
[BUY_SELL] elapsed time: 97442 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 30
[BUY_SELL] elapsed time: 140296 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 40
[BUY_SELL] elapsed time: 192525 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 50
[BUY_SELL] elapsed time: 224899 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 60
[BUY_SELL] elapsed time: 262773 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 70
[BUY_SELL] elapsed time: 325163 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 80
[BUY_SELL] elapsed time: 353672 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 90
[BUY_SELL] elapsed time: 390389 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 100
[BUY_SELL] elapsed time: 421094 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 110
[BUY_SELL] elapsed time: 455000 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 120
[BUY_SELL] elapsed time: 495335 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 130
[BUY_SELL] elapsed time: 507047 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 140
[BUY_SELL] elapsed time: 558155 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 150
[BUY_SELL] elapsed time: 1037757 microseconds
```

```
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[BUY_SELL] THREAD# 20 | elapsed time: 43292 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[BUY_SELL] THREAD# 20 | elapsed time: 75931 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[BUY_SELL] THREAD# 20 | elapsed time: 111277 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[BUY_SELL] THREAD# 20 | elapsed time: 165557 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[BUY_SELL] THREAD# 20 | elapsed time: 186037 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[BUY_SELL] THREAD# 20 | elapsed time: 223723 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[BUY_SELL] THREAD# 20 | elapsed time: 328881 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[BUY_SELL] THREAD# 20 | elapsed time: 300408 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[BUY_SELL] THREAD# 20 | elapsed time: 386391 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[BUY_SELL] THREAD# 20 | elapsed time: 393458 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[BUY_SELL] THREAD# 20 | elapsed time: 435798 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[BUY_SELL] THREAD# 20 | elapsed time: 473112 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[BUY_SELL] THREAD# 20 | elapsed time: 506105 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[BUY_SELL] THREAD# 20 | elapsed time: 541771 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[BUY_SELL] THREAD# 20 | elapsed time: 1027636 microseconds

cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[BUY_SELL] THREAD# 20 | elapsed time: 41654 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[BUY_SELL] THREAD# 20 | elapsed time: 70643 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[BUY_SELL] THREAD# 20 | elapsed time: 110849 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[BUY_SELL] THREAD# 20 | elapsed time: 145546 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[BUY_SELL] THREAD# 20 | elapsed time: 185675 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[BUY_SELL] THREAD# 20 | elapsed time: 226457 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[BUY_SELL] THREAD# 20 | elapsed time: 254579 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[BUY_SELL] THREAD# 20 | elapsed time: 297978 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[BUY_SELL] THREAD# 20 | elapsed time: 325381 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[BUY_SELL] THREAD# 20 | elapsed time: 373709 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[BUY_SELL] THREAD# 20 | elapsed time: 415580 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[BUY_SELL] THREAD# 20 | elapsed time: 457990 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[BUY_SELL] THREAD# 20 | elapsed time: 485298 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[BUY_SELL] THREAD# 20 | elapsed time: 512766 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[BUY_SELL] THREAD# 20 | elapsed time: 1022493 microseconds
```

```
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 10
[RANDOM] elasped time: 47728 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 20
[RANDOM] elasped time: 101580 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 30
[RANDOM] elasped time: 150306 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 40
[RANDOM] elasped time: 166788 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 50
[RANDOM] elasped time: 211816 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 60
[RANDOM] elasped time: 269070 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 70
[RANDOM] elasped time: 319728 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 80
[RANDOM] elasped time: 354105 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 90
[RANDOM] elasped time: 377673 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 100
[RANDOM] elasped time: 470175 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 110
[RANDOM] elasped time: 461965 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 120
[RANDOM] elasped time: 514047 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 130
[RANDOM] elasped time: 586386 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 140
[RANDOM] elasped time: 1021193 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 150
[RANDOM] elasped time: 1041177 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 160
[RANDOM] elasped time: 56751 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 20
[RANDOM] elasped time: 94909 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 30
[RANDOM] elasped time: 136105 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 40
[RANDOM] elasped time: 192669 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 50
[RANDOM] elasped time: 204995 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 60
[RANDOM] elasped time: 233426 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 70
[RANDOM] elasped time: 292255 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 80
[RANDOM] elasped time: 337349 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 90
[RANDOM] elasped time: 413063 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 100
[RANDOM] elasped time: 430648 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 110
[RANDOM] elasped time: 476619 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 120
[RANDOM] elasped time: 520921 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 130
[RANDOM] elasped time: 534953 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 140
[RANDOM] elasped time: 1021212 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 150
[RANDOM] elasped time: 1049199 microseconds
```

```
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[RANDOM] THREAD# 4 | elasped time: 50104 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[RANDOM] THREAD# 4 | elasped time: 98224 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[RANDOM] THREAD# 4 | elasped time: 132522 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[RANDOM] THREAD# 4 | elasped time: 176904 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[RANDOM] THREAD# 4 | elasped time: 222276 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[RANDOM] THREAD# 4 | elasped time: 284760 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[RANDOM] THREAD# 4 | elasped time: 309261 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[RANDOM] THREAD# 4 | elasped time: 334344 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[RANDOM] THREAD# 4 | elasped time: 379903 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[RANDOM] THREAD# 4 | elasped time: 445265 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[RANDOM] THREAD# 4 | elasped time: 458480 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[RANDOM] THREAD# 4 | elasped time: 544257 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[RANDOM] THREAD# 4 | elasped time: 571939 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[RANDOM] THREAD# 4 | elasped time: 650176 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[RANDOM] THREAD# 4 | elasped time: 1035164 microseconds

cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[RANDOM] THREAD# 4 | elasped time: 49293 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[RANDOM] THREAD# 4 | elasped time: 92388 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[RANDOM] THREAD# 4 | elasped time: 138187 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[RANDOM] THREAD# 4 | elasped time: 178490 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[RANDOM] THREAD# 4 | elasped time: 217467 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[RANDOM] THREAD# 4 | elasped time: 256540 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[RANDOM] THREAD# 4 | elasped time: 325434 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[RANDOM] THREAD# 4 | elasped time: 362105 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[RANDOM] THREAD# 4 | elasped time: 396706 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[RANDOM] THREAD# 4 | elasped time: 447874 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[RANDOM] THREAD# 4 | elasped time: 515181 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[RANDOM] THREAD# 4 | elasped time: 534642 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[RANDOM] THREAD# 4 | elasped time: 567817 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[RANDOM] THREAD# 4 | elasped time: 631464 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[RANDOM] THREAD# 4 | elasped time: 1031664 microseconds
```

```
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[RANDOM] THREAD# 20 | elasped time: 38915 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[RANDOM] THREAD# 20 | elasped time: 82814 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[RANDOM] THREAD# 20 | elasped time: 116940 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[RANDOM] THREAD# 20 | elasped time: 159579 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[RANDOM] THREAD# 20 | elasped time: 210150 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[RANDOM] THREAD# 20 | elasped time: 231868 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[RANDOM] THREAD# 20 | elasped time: 275583 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[RANDOM] THREAD# 20 | elasped time: 303188 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[RANDOM] THREAD# 20 | elasped time: 348727 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[RANDOM] THREAD# 20 | elasped time: 378006 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[RANDOM] THREAD# 20 | elasped time: 431145 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[RANDOM] THREAD# 20 | elasped time: 477148 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[RANDOM] THREAD# 20 | elasped time: 498677 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[RANDOM] THREAD# 20 | elasped time: 541085 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[RANDOM] THREAD# 20 | elasped time: 1036747 microseconds

cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[RANDOM] THREAD# 20 | elasped time: 39310 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[RANDOM] THREAD# 20 | elasped time: 74006 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[RANDOM] THREAD# 20 | elasped time: 110663 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[RANDOM] THREAD# 20 | elasped time: 150120 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[RANDOM] THREAD# 20 | elasped time: 187567 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[RANDOM] THREAD# 20 | elasped time: 232692 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[RANDOM] THREAD# 20 | elasped time: 261334 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[RANDOM] THREAD# 20 | elasped time: 309202 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[RANDOM] THREAD# 20 | elasped time: 351405 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[RANDOM] THREAD# 20 | elasped time: 378267 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[RANDOM] THREAD# 20 | elasped time: 426646 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[RANDOM] THREAD# 20 | elasped time: 437432 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[RANDOM] THREAD# 20 | elasped time: 466971 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[RANDOM] THREAD# 20 | elasped time: 516503 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[RANDOM] THREAD# 20 | elasped time: 1026372 microseconds
```

```
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[RANDOM] THREAD# 100 | elasped time: 41222 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[RANDOM] THREAD# 100 | elasped time: 87423 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[RANDOM] THREAD# 100 | elasped time: 122443 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[RANDOM] THREAD# 100 | elasped time: 163085 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[RANDOM] THREAD# 100 | elasped time: 198107 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[RANDOM] THREAD# 100 | elasped time: 229432 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[RANDOM] THREAD# 100 | elasped time: 264345 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[RANDOM] THREAD# 100 | elasped time: 284699 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[RANDOM] THREAD# 100 | elasped time: 338101 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[RANDOM] THREAD# 100 | elasped time: 355541 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[RANDOM] THREAD# 100 | elasped time: 412680 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[RANDOM] THREAD# 100 | elasped time: 451534 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[RANDOM] THREAD# 100 | elasped time: 470334 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[RANDOM] THREAD# 100 | elasped time: 507338 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[RANDOM] THREAD# 100 | elasped time: 1059227 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[RANDOM] THREAD# 100 | elasped time: 41005 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[RANDOM] THREAD# 100 | elasped time: 73642 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[RANDOM] THREAD# 100 | elasped time: 107717 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[RANDOM] THREAD# 100 | elasped time: 144072 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[RANDOM] THREAD# 100 | elasped time: 186371 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[RANDOM] THREAD# 100 | elasped time: 207799 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[RANDOM] THREAD# 100 | elasped time: 264548 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[RANDOM] THREAD# 100 | elasped time: 287063 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[RANDOM] THREAD# 100 | elasped time: 318310 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[RANDOM] THREAD# 100 | elasped time: 348598 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[RANDOM] THREAD# 100 | elasped time: 381374 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[RANDOM] THREAD# 100 | elasped time: 425869 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[RANDOM] THREAD# 100 | elasped time: 464850 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[RANDOM] THREAD# 100 | elasped time: 527661 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[RANDOM] THREAD# 100 | elasped time: 1029149 microseconds
```

```
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 10
[SHOW] elasped time: 45641 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 20
[SHOW] elasped time: 106021 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 30
[SHOW] elasped time: 160238 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 40
[SHOW] elasped time: 182203 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 50
[SHOW] elasped time: 221057 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 60
[SHOW] elasped time: 272099 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 70
[SHOW] elasped time: 332403 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 80
[SHOW] elasped time: 339240 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 90
[SHOW] elasped time: 389110 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 100
[SHOW] elasped time: 410474 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 110
[SHOW] elasped time: 447785 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 120
[SHOW] elasped time: 524889 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 130
[SHOW] elasped time: 558550 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 140
[SHOW] elasped time: 1018762 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 150
[SHOW] elasped time: 1054088 microseconds

cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 10
[SHOW] elasped time: 49815 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 20
[SHOW] elasped time: 97573 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 30
[SHOW] elasped time: 140607 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 40
[SHOW] elasped time: 177509 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 50
[SHOW] elasped time: 252448 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 60
[SHOW] elasped time: 249916 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 70
[SHOW] elasped time: 340166 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 80
[SHOW] elasped time: 334759 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 90
[SHOW] elasped time: 361874 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 100
[SHOW] elasped time: 440550 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 110
[SHOW] elasped time: 477315 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 120
[SHOW] elasped time: 518650 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 130
[SHOW] elasped time: 556538 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 140
[SHOW] elasped time: 1024534 microseconds
cse20160169@cspro8:~/p2/project1$ ./multiclient 172.30.10.11 60073 150
[SHOW] elasped time: 1055943 microseconds
```

```
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[SHOW] THREAD# 20 | elasped time: 45670 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[SHOW] THREAD# 20 | elasped time: 79231 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[SHOW] THREAD# 20 | elasped time: 123864 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[SHOW] THREAD# 20 | elasped time: 168656 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[SHOW] THREAD# 20 | elasped time: 201724 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[SHOW] THREAD# 20 | elasped time: 232181 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70\
> ^C
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[SHOW] THREAD# 20 | elasped time: 273168 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[SHOW] THREAD# 20 | elasped time: 303800 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[SHOW] THREAD# 20 | elasped time: 355260 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[SHOW] THREAD# 20 | elasped time: 387711 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[SHOW] THREAD# 20 | elasped time: 404975 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[SHOW] THREAD# 20 | elasped time: 466377 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[SHOW] THREAD# 20 | elasped time: 496251 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[SHOW] THREAD# 20 | elasped time: 1038091 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[SHOW] THREAD# 20 | elasped time: 1067718 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 10
[SHOW] THREAD# 20 | elasped time: 45254 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 20
[SHOW] THREAD# 20 | elasped time: 82124 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 30
[SHOW] THREAD# 20 | elasped time: 118023 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 40
[SHOW] THREAD# 20 | elasped time: 157890 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 50
[SHOW] THREAD# 20 | elasped time: 189138 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 60
[SHOW] THREAD# 20 | elasped time: 232372 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 70
[SHOW] THREAD# 20 | elasped time: 271362 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 80
[SHOW] THREAD# 20 | elasped time: 319732 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 90
[SHOW] THREAD# 20 | elasped time: 327800 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 100
[SHOW] THREAD# 20 | elasped time: 367298 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 110
[SHOW] THREAD# 20 | elasped time: 417505 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 120
[SHOW] THREAD# 20 | elasped time: 446268 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 130
[SHOW] THREAD# 20 | elasped time: 473785 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 140
[SHOW] THREAD# 20 | elasped time: 511434 microseconds
cse20160169@cspro8:~/p2/project2$ ./multiclient 172.30.10.11 60073 150
[SHOW] THREAD# 20 | elasped time: 1032161 microseconds
```