

System Programming Project 3

담당 교수 : 이영민 교수님

이름 : 신영원

학번 : 20211546

1. 개발 목표

서버의 역할은 client로부터 "show", "buy", "sell", "exit" 명령을 받아, 이진 탐색 트리를 기반으로 stock 정보를 조회하고 수정하고, 이 내용을 stock.txt파일에 기록하는 것이다.

또한, 이를 다수의 동시 접속 클라이언트의 요청을 수행하는 것이 서버의 특징이다.

2. 개발 범위 및 내용

A. 개발 범위

- 아래 항목을 구현했을 때의 결과를 간략히 서술

1. Task 1: Event-driven Approach

-단일 프로세스, 단일 스레드 내에서 select()를 이용하여 여러 소켓을 감시하고, 100인의 클라이언트까지 커버 가능하다. 추가

2. Task 2: Thread-based Approach

- accept로 접속을 수락하고 fd를 공유 버퍼에 삽입한다. IO 대기 도중에 다른 client를 커버할 수 있고,이를 통해 cpu 코어를 효과적으로 활용할 수 있다.

3. Task 3: Performance Evaluation

-모든 자식 프로세스가 exit된 이후에, 프로그램 시작 직전의 시각과 종료 시각의 차이만큼을 초와 마이크로초 단위로 출력한다.

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: fd_set 집합을 통해 listen fd와 모든 client fd를 관리한다. 읽을 수 있는 소켓에 대해 Rio_readlineb를 호출하여 명령을 처리한다.

✓ epoll과의 차이점 서술: epoll은 등록된 fd 변경만 커널 내부에서 관리하지만

규모가 큰 동시 접속 환경에서 효율적이다. Select는 fd 의 수에 제한이 있으며 호출마다 fd비트맵을 복사한다.

- **Task2 (Thread-based Approach with pthread)**

- ✓ Master Thread의 Connection 관리: listenfd에 새 연결이 오면 accept하고 반환된 connfd를 sbuf_insert에 인자로 넘긴다. 그리고 공유 버퍼에 빈 슬롯이 없으면 대기한다.

- **Task3 (Performance Evaluation)**

- ✓ 얻고자 하는 metric 정의, 그렇게 정한 이유, 측정 방법 서술

-elapsed time: 모든 client process가 exit되기까지의 소요 시간

-이유: 모든 client의 요청이 처리되는 시간을 보이는 게 직관적이고, 시스템 전체의 처리 capacity를 확인할 수 있는 지표로 활용할 수 있다고 생각하였다.

-측정방법: multicient실행 직전과 직후에 gettimeofday()를 호출하여 elapsed time 측정 가능.

- ✓ Configuration 변화에 따른 예상 결과 서술

-event-driven approach 든 thread-based approach 든 show, sell, buy 의 수행속도 차이는 거의 일정할 것이다. (show 가 sell/buy 보다 시간이 덜 소요될 것)

-event-driven approach 는 thread-based approach 보다 시간이 덜 소요될 것.

C. 개발 방법

Event-driven approach:

-client배열을 활용하여 수락된 각 클라이언트 소켓 디스크립터를 관리하고, 각 클라이언트마 rio버퍼를 관리할 수 있게끔 rio_arr배열을 활용한다.

-모든 소켓 집합과 검사용 집합 fd_set 변수 2개를 설정하고, 리스닝 소켓을 등록한다. 그리고 select를 호출할 때 maxfd+1을 인자로 보낸다.

-파일에서 stock를 읽어 이진 탐색 트리에 저장한다. 그리고 수신용 소켓을 열고 fd_set에 추가한다. 그리고 client배열을 초기화해준다.

-select가 반환되었을 때, 리스닝 소켓이 준비된 경우, accept로 연결을 수락하고, 빈 슬롯을 찾아 그 소켓 디스크립터를 저장한다. 그 뒤로, rio 버퍼를 초기화하고 fd_set에 추가하며 maxfd를 갱신한다. Client 소켓이 준비된 경우에는 rio 라인 읽기 함수로부터 한 줄씩 수신하고, 명령 텍스트를 파싱해서 이진 탐색 트리 관리 함수를 호출한다.

Thread-based approach:

-master thread의 경우, 공유 버퍼 초기화 함수를 호출하고 원하는 worker thread 수만큼 pthread_create로 thread 풀을 생성하고, 스레드가 시작하는 즉시 pthread_detach로 처리한다.

-worker thread의 경우, 각 worker는 sbuf_remove를 호출하여 master로부터 디스크립터를 꺼내고, 꺼낸 디스크립터로 rio 라인을 읽으며 수행한다. 종료나 exit명령이 오면 소켓을 닫고 루프로 회귀한다.

3. 구현 결과

Event-driven approach: 단일 스레드로 최대 100인의 client의 연결을 커버할 수 있음

Thread-based approach: 다수의 worker thread로 client를 병렬적으로 처리할 수 있음.

4. 성능 평가 결과 (Task 3)

- 강의자료 슬라이드의 내용 참고하여 작성 (측정 시점, 출력 결과 값 캡처 포함)

- Event-driven approach, thread-based approach 각각 5가지의 사례를 살펴보며 분석을 진행한다.

- <Event-driven approach>

1)buy, sell, show(순서대로 10명, 20명일 때)

[sell] success	buy 4 7
buy 1 4	[buy] success
[buy] success	buy 3 2
sell 1 1	[buy] success
[sell] success	sell 10 2
buy 9 1	[sell] success
[buy] success	buy 1 5
buy 10 2	[buy] success
[buy] success	sell 3 5
buy 9 2	[sell] success
[buy] success	buy 1 10
sell 6 2	[buy] success
[sell] success	sell 5 3
show	[sell] success
1 132 1000	buy 10 3
2 38 500	[buy] success
3 65 700	show
4 118 800	1 116 1000
5 184 1200	2 87 500
6 14 1100	3 80 700
7 9 950	4 114 800
8 67 1050	5 166 1200
9 83 1150	6 8 1100
10 117 900	7 2 950
Elapsed time: 10.064981 seconds	8 50 1050
	9 57 1150
	10 104 900
	Elapsed time: 10.079067 seconds

- 2)buy, sell(순서대로 10명, 20명일 때)

buy 5 3	sell 7 3
[buy] success	[sell] success
buy 5 1	buy 6 7
[buy] success	[buy] success
sell 5 3	buy 5 7
[sell] success	[buy] success
sell 2 5	sell 8 2
[sell] success	[sell] success
buy 5 4	sell 7 1
[buy] success	[sell] success
sell 10 9	sell 2 5
[sell] success	[sell] success
sell 10 5	buy 2 7
[sell] success	[buy] success
buy 3 1	sell 6 10
[buy] success	[sell] success
buy 4 10	buy 5 6
[buy] success	[buy] success
buy 6 2	sell 8 9
[buy] success	[sell] success
sell 8 4	sell 4 9
[sell] success	[sell] success
sell 5 5	sell 1 5
[sell] success	[sell] success
sell 2 7	sell 1 6
[sell] success	[sell] success
buy 5 10	buy 4 8
[buy] success	[buy] success
Elapsed time: 8.039370 seconds	Elapsed time: 9.048901 seconds

- 3)buy, show(순서대로 10명, 20명일 때)

```

buy 9 9
[buy] success
buy 3 8
[buy] success
show
1 152 1000
2 84 500
3 24 700
4 90 800
5 181 1200
6 0 1100
7 0 950
8 22 1050
9 56 1150
10 125 900
show
1 152 1000
2 84 500
3 24 700
4 90 800
5 181 1200
6 0 1100
7 0 950
8 22 1050
9 56 1150
10 125 900
buy 7 3
Not enough left stock
Elapsed time: 8.037006 seconds

```

```

7 0 950
8 0 1050
9 38 1150
10 82 900
buy 1 7
[buy] success
show
1 75 1000
2 60 500
3 6 700
4 72 800
5 169 1200
6 0 1100
7 0 950
8 0 1050
9 38 1150
10 82 900
show
1 75 1000
2 60 500
3 6 700
4 72 800
5 169 1200
6 0 1100
7 0 950
8 0 1050
9 38 1150
10 82 900
Elapsed time: 8.043749 seconds

```

- 4)sell(순서대로 10명, 20명일 때)

```

sell 9 4
[sell] success
sell 7 8
[sell] success
sell 6 7
[sell] success
sell 2 9
[sell] success
sell 2 10
[sell] success
sell 1 5
[sell] success
sell 2 4
[sell] success
sell 2 1
[sell] success
sell 10 10
[sell] success
sell 10 7
[sell] success
sell 2 8
[sell] success
sell 5 10
[sell] success
sell 4 10
[sell] success
sell 1 6
[sell] success
Elapsed time: 5.038717 second:

```

```

sell 8 7
[sell] success
sell 10 9
[sell] success
sell 6 10
[sell] success
sell 7 9
[sell] success
sell 1 1
[sell] success
sell 2 5
[sell] success
sell 7 6
[sell] success
sell 5 8
[sell] success
sell 3 1
[sell] success
sell 7 9
[sell] success
sell 5 6
[sell] success
sell 10 9
[sell] success
sell 5 7
[sell] success
sell 4 5
[sell] success
Elapsed time: 6.057526 seconds

```

- 5)show(순서대로 10명, 20명일 때)

5 236 1200	5 236 1200
6 88 1100	6 88 1100
7 75 950	7 75 950
8 83 1050	8 83 1050
9 62 1150	9 62 1150
10 149 900	10 149 900
show	show
1 97 1000	1 97 1000
2 138 500	2 138 500
3 30 700	3 30 700
4 115 800	4 115 800
5 236 1200	5 236 1200
6 88 1100	6 88 1100
7 75 950	7 75 950
8 83 1050	8 83 1050
9 62 1150	9 62 1150
10 149 900	10 149 900
show	show
1 97 1000	1 97 1000
2 138 500	2 138 500
3 30 700	3 30 700
4 115 800	4 115 800
5 236 1200	5 236 1200
6 88 1100	6 88 1100
7 75 950	7 75 950
8 83 1050	8 83 1050
9 62 1150	9 62 1150
10 149 900	10 149 900

- Elapsed time: 5.003002 seconds Elapsed time: 7.007956 seconds

<Thread-based approach>

1) buy, sell, show(순서대로 10명, 20명)

9 109 1000	sell 7 2
10 6 1000	[sell] success
sell 8 8	sell 10 7
[sell] success	[sell] success
show	show
1 109 1000	1 99 1000
2 84 1000	2 41 1000
3 55 3000	3 88 3000
4 423 4000	4 433 4000
5 134 3000	5 148 3000
6 98 2000	6 110 2000
7 83 1000	7 57 1000
8 75 1000	8 62 1000
9 189 1000	9 181 1000
10 6 1000	10 18 1000
buy 6 2	show
[buy] success	1 99 1000
show	2 41 1000
1 109 1000	3 88 3000
2 84 1000	4 433 4000
3 55 3000	5 148 3000
4 423 4000	6 110 2000
5 134 3000	7 57 1000
6 96 2000	8 62 1000
7 83 1000	9 181 1000
8 75 1000	10 18 1000
9 189 1000	buy 8 5
10 6 1000	[buy] success

Elapsed time: 20.046369 seconds Elapsed time: 30.086696 seconds

2) buy, sell(순서대로 10명, 20명)

sell 5 5	sell 1 7
[sell] success	[sell] success
buy 1 5	buy 4 4
[buy] success	[buy] success
buy 6 9	sell 5 5
[buy] success	[sell] success
buy 3 7	buy 5 6
[buy] success	[buy] success
sell 1 5	sell 2 10
[sell] success	[sell] success
sell 7 4	sell 8 10
[sell] success	[sell] success
sell 4 4	sell 10 9
[sell] success	[sell] success
buy 3 9	buy 6 10
[buy] success	[buy] success
buy 4 2	sell 1 10
[buy] success	[sell] success
sell 1 7	sell 5 6
[sell] success	[sell] success
sell 9 1	sell 5 2
[sell] success	[sell] success
buy 7 1	sell 8 10
[buy] success	[sell] success
buy 10 6	sell 4 7
[buy] success	[sell] success
buy 9 1	sell 6 2
[buy] success	[sell] success

Elapsed time: 12.041097 seconds Elapsed time: 21.073655 seconds

3) buy, show(순서대로 10명, 20명)

```
9 136 1000          9 81 1000
10 1 1000           10 1 1000
show                buy 4 2
1 131 1000          [buy] success
2 0 1000            show
3 108 3000          1 94 1000
4 469 4000          2 0 1000
5 153 3000          3 51 3000
6 135 2000          4 420 4000
7 60 1000           5 108 3000
8 27 1000           6 95 2000
9 136 1000          7 31 1000
10 1 1000            8 1 1000
show                9 81 1000
1 131 1000          10 1 1000
2 0 1000            buy 3 6
3 108 3000          [buy] success
4 469 4000          show
5 153 3000          1 94 1000
6 135 2000          2 0 1000
7 60 1000           3 45 3000
8 27 1000           4 420 4000
9 136 1000          5 108 3000
10 1 1000           6 95 2000
buy 3 7             7 31 1000
[buy] success       8 1 1000
buy 7 2             9 81 1000
[buy] success       10 1 1000
Elapsed time: 14.034013 seconds Elapsed time: 21.042972 seconds
```

4) sell(순서대로 10명, 20명)

```
sell 4 5            sell 10 5
[sell] success      [sell] success
sell 7 6            sell 7 6
[sell] success      [sell] success
sell 10 8           sell 9 3
[sell] success      [sell] success
sell 9 6            sell 8 2
[sell] success      [sell] success
sell 7 9            sell 5 8
[sell] success      [sell] success
sell 2 6            sell 3 10
[sell] success      [sell] success
sell 4 7            sell 3 3
[sell] success      [sell] success
sell 10 9           sell 5 5
[sell] success      [sell] success
sell 5 3            sell 6 1
[sell] success      [sell] success
sell 1 2            sell 10 8
[sell] success      [sell] success
sell 6 1            sell 10 9
[sell] success      [sell] success
sell 7 9            sell 7 9
[sell] success      [sell] success
sell 4 10           sell 6 4
[sell] success      [sell] success
sell 4 7            sell 10 2
[sell] success      [sell] success
Elapsed time: 5.047226 seconds Elapsed time: 11.094440 seconds
```

5) show(순서대로 10명, 20명)

5 151 3000	5 151 3000
6 152 2000	6 152 2000
7 109 1000	7 109 1000
8 41 1000	8 41 1000
9 117 1000	9 117 1000
10 88 1000	10 88 1000
show	show
1 142 1000	1 142 1000
2 34 1000	2 34 1000
3 101 3000	3 101 3000
4 488 4000	4 488 4000
5 151 3000	5 151 3000
6 152 2000	6 152 2000
7 109 1000	7 109 1000
8 41 1000	8 41 1000
9 117 1000	9 117 1000
10 88 1000	10 88 1000
show	show
1 142 1000	1 142 1000
2 34 1000	2 34 1000
3 101 3000	3 101 3000
4 488 4000	4 488 4000
5 151 3000	5 151 3000
6 152 2000	6 152 2000
7 109 1000	7 109 1000
8 41 1000	8 41 1000
9 117 1000	9 117 1000
10 88 1000	10 88 1000
Elapsed time: 9.005029 seconds	Elapsed time: 10.004138 seconds

분석:

show는 단순히 메모리 순회와 네트워크 송신만 하는 반면, buy나 sell은 그 뒤에 persist_stock을 통해 파일 전체를 덤프하기에 오버헤드가 추가로 발생한다.

그렇기에 show와 buy, sell의 조합에 따라 client가 10명에서 20명이 될 때, 증가하는 소요 시간이 차이를 보이는 것이다.