

자료구조 실습과제 07

- 개발 도구 : visual studio를 이용한 C프로그래밍
- 솔루션 명칭 : Sol_07_이름이니셜(예: Sol_07_hsoh)
- 프로젝트 명칭 : 문제 번호(예: Prob_01)
- 제출방법 : 아래 문제를 하나의 솔루션에 여러 개의 프로젝트로 구성하여 프로그램한 후, 컴파일 및 실행해보고 오류가 없으면, 메뉴에서 솔루션 정리를 수행한 후 윈도우 탐색기에 서 솔루션 폴더를 찾아 압축하여 E-class에 올림

문제 1) 큐 자료구조를 이용하여 큐잉이론에 따라 문제 특성을 분석하여 시뮬레이션하여 분석 결과를 제시하는 프로그램을 작성하시오.

문제 상황 : 햄버거 가게에서 두 사람이 햄버거를 만들어 서비스 제공한다. 판매한 햄버거 종류별 개수와 해당 서비스를 하는 데 총 걸린 시간, 햄버거별 서비스를 제공하는데 걸린 평균시간, 주문이 취소된 건 수를 주어진 상황에서 분석하는 프로그램을 작성하시오.

- 주문은 매 5 마다 손님이 방문하여 주문을 한다. 주문할 수 있는 내용은 아래 표와 같다.

메뉴	소요시간
MENU 0	5
MENU 1	10
MENU 2	15
MENU 3	20
MENU 4	25

- 손님은 $\text{rand()} \% 5$ 로 메뉴를 결정하고, 주문한다. 주문된 내용은 Queue에 저장된다. 주문 즉시 햄버거가 만들어지지 않고, 주문이 Queue에 들어가기 때문에 버거를 만드는 시간보다 손님은 더 기다릴 수 있다.
- Queue가 가득 찬 경우에 손님이 주문할 경우 주문을 넣을 수 없어 해당 손님은 주문을 하지 않고 떠난다.
- 주문을 처리하는 두 사람은 현재 처리하는 작업이 없으면 바로 Queue로부터 다음 주문을 추출하여 처리한다.
- 하나의 서비스가 완료되면, 손님이 주문을 한 시간부터 서비스가 완료되기까지의 시간을 구하여 대기시간으로 한다.
- 이러한 서비스를 WORK-TIME 1000 동안 수행할 경우 메뉴별 서비스한 개수, 누적대기시간, 평균서비스시간과 취소된 주문의 건 수를 분석하시오.(단, 메뉴 선택 패턴이 동일하게 하고, 큐의 크기를 10, 20, 30, 40, 50일 때 처리되는 주문 건 수와 각 손님의 평균 대기 시간을 구하여 분석하시오.

실행 예

큐 의 크 기 가 10 일 때	<pre> ===== Results ===== MENU 0 24개 서비스, 누적대기시간 1520, 평균서비스시간 63.333333 MENU 1 26개 서비스, 누적대기시간 1985, 평균서비스시간 76.346154 MENU 2 31개 서비스, 누적대기시간 2525, 평균서비스시간 81.451613 MENU 3 19개 서비스, 누적대기시간 1560, 평균서비스시간 82.105263 MENU 4 30개 서비스, 누적대기시간 2450, 평균서비스시간 81.666667 취소된 주문 60 개 ===== </pre>
큐 의 크 기 가 20 일 때	<pre> ===== Results ===== MENU 0 28개 서비스, 누적대기시간 3240, 평균서비스시간 115.714286 MENU 1 30개 서비스, 누적대기시간 3740, 평균서비스시간 124.666667 MENU 2 24개 서비스, 누적대기시간 3375, 평균서비스시간 140.625000 MENU 3 20개 서비스, 누적대기시간 2625, 평균서비스시간 131.250000 MENU 4 30개 서비스, 누적대기시간 3735, 평균서비스시간 124.500000 취소된 주문 46 개 ===== </pre>
큐 의 크 기 가 10 일 때	<pre> ===== Results ===== MENU 0 30개 서비스, 누적대기시간 4500, 평균서비스시간 150.000000 MENU 1 27개 서비스, 누적대기시간 4120, 평균서비스시간 152.592593 MENU 2 29개 서비스, 누적대기시간 5445, 평균서비스시간 187.758621 MENU 3 22개 서비스, 누적대기시간 3630, 평균서비스시간 165.000000 MENU 4 26개 서비스, 누적대기시간 3830, 평균서비스시간 147.307692 취소된 주문 34 개 ===== </pre>
큐 의 크 기 가 30 일 때	<pre> ===== Results ===== MENU 0 28개 서비스, 누적대기시간 4480, 평균서비스시간 160.000000 MENU 1 28개 서비스, 누적대기시간 4835, 평균서비스시간 172.678571 MENU 2 29개 서비스, 누적대기시간 6150, 평균서비스시간 212.068966 MENU 3 22개 서비스, 누적대기시간 3940, 평균서비스시간 179.090909 MENU 4 27개 서비스, 누적대기시간 4460, 평균서비스시간 165.185185 취소된 주문 26 개 ===== </pre>
큐 의 크 기 가 40 일 때	<pre> ===== Results ===== MENU 0 26개 서비스, 누적대기시간 3940, 평균서비스시간 151.538462 MENU 1 28개 서비스, 누적대기시간 4900, 평균서비스시간 175.000000 MENU 2 28개 서비스, 누적대기시간 5900, 평균서비스시간 210.714286 MENU 3 22개 서비스, 누적대기시간 3970, 평균서비스시간 180.454545 MENU 4 28개 서비스, 누적대기시간 4805, 평균서비스시간 171.607143 취소된 주문 17 개 ===== </pre>

프로그램의 메인 루프 구조 : 아래 코드 참조

```
while (time < WORK_TIME) { // time을 1씩 증가시키면서
    if (time % 5 == 0) { // 새로운 손님 도착하여 주문
        //주문을 생성하여 큐에 넣기
    }

    if ( !isBusyServer1 ) { //놓고 있다가 새로운 서비스 시작
        ...
    }
    else if (isBusyServer1 && server1.processing_time == 0) { //작업 끝남.
        ...
    }

    if (!isBusyServer2) { //놓고 있다가 새로운 서비스 시작
        ...
    }
    else if (isBusyServer2 && server2.processing_time == 0) { //작업 끝남.
        ...
    }

    // 시간 업데이트
    time++;
    // server1의 시간 1 감소
    // server2의 시간 1 감소
}

// 결과 출력
printf("===== Results =====\n");
for (i = 0; i < 5; i++) {
    printf("MENU %d %d개 서비스, 누적대기시간 %d, 평균서비스시간 %f\n", ...}
    printf("취소된 주문 %d 개\n", q.lost_service);
    printf("===== \n");
}
```