

객체지향 프로그래밍 과제 보고서

202104389 황형진, 202104309 이진용

1. 문제 정의

이번 문제는 컴퓨터의 메모리 장치를 모델링하는 Ram 클래스를 구현하는 과제입니다. 이 클래스는 컴퓨터 메모리의 일부를 시뮬레이션하여, 특정 주소에 데이터를 기록하고, 그 데이터를 다시 읽어오는 기능을 제공합니다. 메모리는 고정된 크기(100KB)로 할당되며, 각 메모리 주소는 1바이트 크기의 데이터를 저장할 수 있습니다. 이를 통해 기본적인 메모리의 읽기 및 쓰기 작업을 수행할 수 있는 메모리 관리 시스템을 설계하고 구현하는 것이 목표입니다.

2. 문제 해결 방법

1. 메모리 할당 : 메모리 크기는 100KB로 고정되어 있습니다. 따라서 char 배열을 사용하여 100KB의 메모리 공간을 확보하고, 각 배열 인덱스를 메모리 주소로 사용하도록 설정합니다.
2. 데이터 읽기 및 쓰기 : 주어진 주소에 데이터를 저장하고, 그 주소로부터 데이터를 읽어오는 기능을 구현하기 위해, 배열 인덱스를 이용하여 데이터의 기록과 읽기 작업을 수행합니다. 이때 write 함수는 주어진 주소에 1바이트의 데이터를 기록하며, read 함수는 주어진 주소에서 1바이트의 데이터를 반환하도록 설계되었습니다.
3. 메모리 초기화 : Ram 객체가 생성될 때 메모리 공간을 모두 0으로 초기화하여, 처음 사용할 때 불필요한 데이터를 없앱니다.
4. 소멸자 활용 : 프로그램 종료 시 메모리가 정상적으로 해제되었는지 확인하기 위해 소멸자(~Ram())를 정의하여, Ram 객체가 소멸될 때 "메모리 제거됨"이라는 메시지를 출력하도록 설계합니다.

3. 아이디어 평가

1. 메모리 할당 : 100KB 메모리를 할당하는 방식은 배열을 사용하여 간단하게 처리할 수 있었습니다. C++에서 char 타입을 사용하면 메모리의 1바이트 단위로 데이터를 관리할 수 있기 때문에 적합했습니다.
2. 데이터 읽기 및 쓰기 : write 및 read 함수는 배열 인덱스를 사용하여 주소 기반 메모리 관리를 수행했으며, 각각 1바이트 단위로 데이터를 읽고 쓰는 기능이 제대로 작동했습니다. 이 방식은 효율적이고, 실제 메모리 관리 동작을 시뮬레이션하는 데 문제 없이 작동했습니다.
3. 메모리 초기화 : 메모리 초기화 작업을 통해 불필요한 값들이 배열에 남아 있지 않도록 했으며, 이로 인해 데이터를 읽고 쓸 때 예상치 못한 오류가 발생하지 않았습니다.
4. 소멸자 활용 : 프로그램 종료 시 객체의 소멸자가 호출되어 "메모리 제거됨" 메시지가 출력되는 것을 확인할 수 있었습니다. 이를 통해 Ram 객체가 정상적으로 소멸되는 과정을 확인할 수 있었습니다.

4. 문제를 해결한 키 아이디어 또는 알고리즘 설명

문제를 해결하는 핵심은 배열을 메모리처럼 사용하여 데이터를 주소 기반으로 기록하고 읽어오는 구조를 구현한 것입니다. 다음과 같이 4가지 사항으로 정리할 수 있습니다.

1. 메모리 시뮬레이션 : Ram 클래스는 char 배열을 사용하여 100KB 메모리 공간을 할당합니다. 이 배열은 실제 메모리 주소와 같이 동작하며, 배열 인덱스가 주소 역할을 합니다.
2. 데이터 기록 (write) : write(int address, char value) 함수는 배열의 특정 인덱스(주소)에 1바이트 데이터를 저장합니다. 이때 address는 배열의 인덱스에 해당하며, 저장할 데이터는 value로 전달됩니다.
3. 데이터 읽기 (read) : read(int address) 함수는 배열의 특정 인덱스에서 데이터를 읽어옵니다. 이때도 address는 배열의 인덱스 역할을 하며, 해당 인덱스에 저장된 데이터(1바이트)를 반환합니다.
4. 소멸자 : ~Ram() 소멸자는 객체가 소멸될 때 호출되며, 메모리 해제 작업을 시뮬레이션하는 역할을 합니다. 이를 통해 메모리가 제대로 해제되었음을 사용자에게 알려줍니다.

이번 과제를 통해 메모리 시스템을 간단하게 시뮬레이션할 수 있도록 설계되었으며, 기본적인 메모리 관리 원리를 이해하는 데 도움이 되었습니다. 이를 통해 실제 메모리에서 데이터를 저장하고 가져오는 과정을 코드를 통해 파악할 수 있었습니다. 또한 헤더파일과 클래스.cpp파일, 메인파일을 분리해 코드를 짜는 법을 학습해 객체지향 프로그래밍의 원리를 이해하는데 많은 도움이 되었습니다.

실행 결과

