

```
#include<iostream>
```

```
#include<fstream>
```

```
using namespace std;
```

```
int Load() {
```

```
    ifstream instream;
```

```
    instream.open("input.txt");
```

```
    int num;
```

```
    while (!instream.eof())
```

```
        instream >> num;
```

```
    instream.close();
```

```
    return num;
```

```
}
```

```
void Store(int data) {
```

```
    ofstream outstream;
```

```
    outstream.open("input.txt", ios::app);
```

```
    outstream << data << endl;
```

```
    outstream.close();
```

```
}
```

```
int Add(int i, int j) {
```

```
    return i + j
```

```
}
```

```
// 데이터파일의open, close는Load, Store 각함수안에서매번수행하게할것
```

```

int main() {
for (int i = 0; i < 1000; ++i) {
int x = Load();
x = Add(x, 1);
Store(x);
}
return 0;
}

```

1)  $n$ 개를 동시에 실행했을 때 데이터 파일의 마지막 숫자가  $1000*n$ 인가?

아니다.

2) 결과가  $1000*n$ 이 아니라면 그 원인은 무엇인지 설명하시오. (데이터 파일 중 문제가 발생한 부분을 10줄 이내로 삽입)

맨 처음 작동 시킨 프로세스를 p1 이라고 한다. 두 번째로 작동 시킨 프로세스 p2 라고 한다. 윈도우는 프로그램 스케줄에 따라 time sharing을 한다. 그렇다면 p1 이 작동되는 중간에 p2가 실행됨에 따라 프로그램 중간 중간에 interrupt가 발생한다. 그렇기 때문에 interrupt가 발생함에 따라 최종값이  $1000*n$  보다 작게 나온다.

또한 데이터 파일의 경우 하드디스크에 위치해있고 p1,p2 는 ram에 위치해 있는데, 만약 하드디스크에 쓰는 도중에 p1이 ram에 있어도 속도는 ram의 속도가 아닌 하드디스크의 속도에 맞춰지는데 그 사이에 p2가 실행된다면 문제가 발생하기 때문에 하드디스크와 ram의 쿨럭차이에 의해서도 문제가 발생할 수 있다. 이 문제로 인해 문제가 되는 부분과 결과 값은 항상 유동적이다.

열기(0) ▾



저장(S)

63

64

65

66

67

68

69

70

71

72

73

74

75

76

76

77

78

78

79

79

...

일반 텍스트 ▾

탭 너비: 8 ▾

3001행, 5열 ▾

삽입