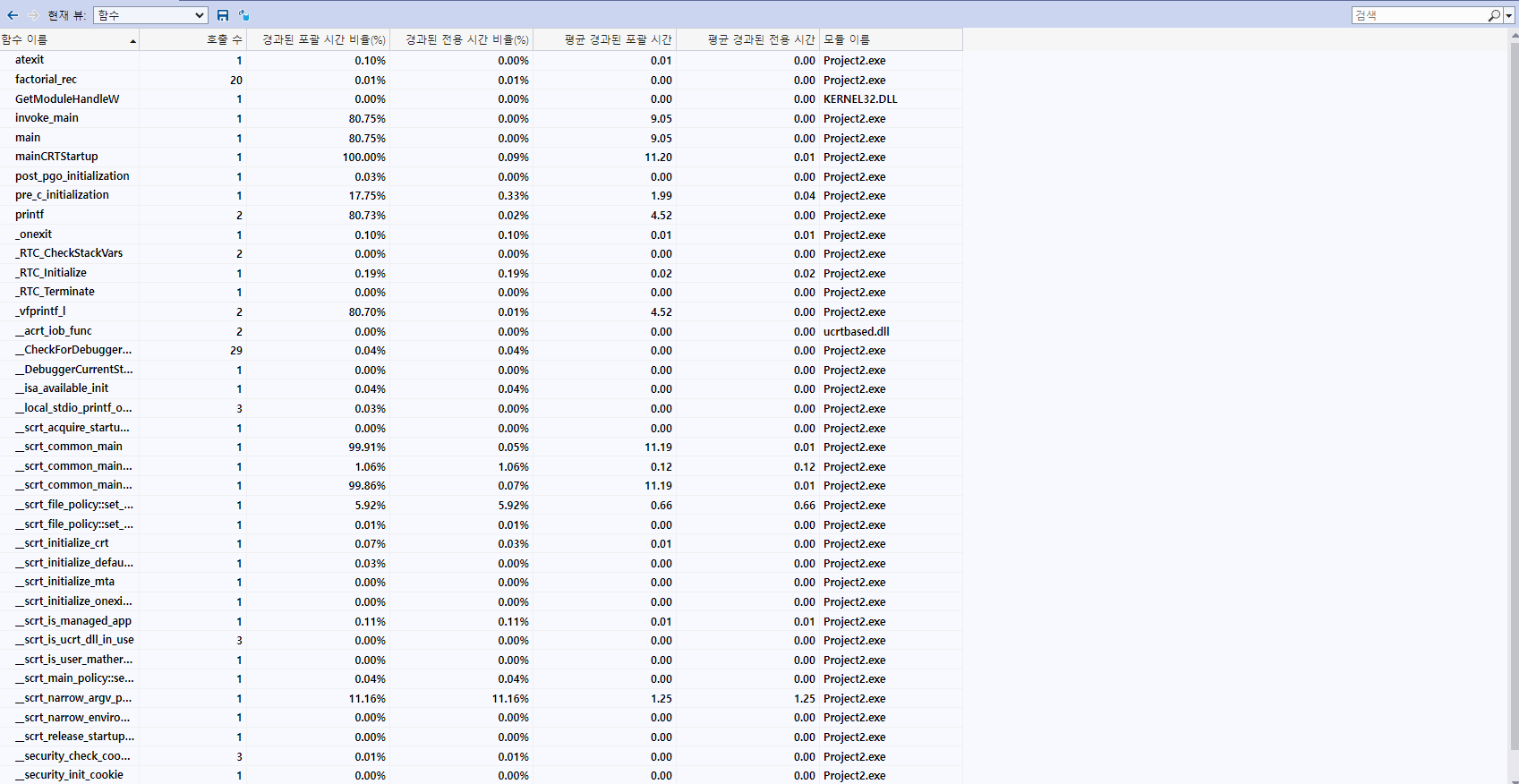
iteration 함수와 recursion 함수의 결과와 과정

5846593 이지현

20!를 iteration(반복) 함수와 recursion(순환) 함수로 표현하여 성능 프로파일러를 사용한 결과입니다.



factorial\_iter의 호출 수와 시간 비율이 나오지 않아 중단점을 사용하여 시간과 호출 수 비교를 하였습니다.

테이블이(가) 표시된 사진

자동 생성된 설명

줄 3이 반복 함수입니다. 반복 함수를 사용한 경우에는 1번 호출되었으며 0.03초 걸렸습니다. 줄 12는 순환 함수입니다. 순환함수를 사용한 경우에는 20번 호출되었으며 0.6초가 걸렸습니다.

반복함수는 반복문을 통해 일정정해진 구간을 반복해서 문제를 해결하는 방식이기때문에 반복이 순환보다 수행속도가 상대적을 빠릅니다. 순환 함수는 함수가 수행도중 자기자신을 다시 호출하여 문제를 해결하는 방식입니다. 순환함수는 호출하기 위해서는 함수의 매개변수들을 스택에 저장하는 것과 같은 사전 작업이 필요합니다. 이 때문에 반복함수보다 순환함수가 수행시간이 더 걸린다.

예로 3!로 과정을 설명해 보겠습니다.

double factorial\_iter(double n) {

double result = 1;

if (n <= 1) return 1;

for (double i = 2; i <= n; i++) {

result = result \* i;

}

return(result);

}

반복함수를 사용했을 시에는 변수 하나를 선언하고 반복문을 사용해 변수 값에 3부터 1까지 값을 곱해 3!의 결과 값이 도출됩니다.

double factorial\_rec(double n) {

if (n <= 1)return(1);

else return(n \* factorial\_rec(n - 1));

}

순환함수를 사용했을 시에는 factorial(3)을 수행하는 도중에 factorial(2)를 호출하게 되고 factorial(2)는 factorial(1)을 호출하게 됩니다. factorial(1)은 1을 반환하게 되고 그 반환 (값이 factorial(2)로 전달되고 factorial(2)의 반환 값이 factorial(3)로 전달되어 3!의 결과 값이 도출됩니다.

이를 통해 순환은 상대적으로 가독성은 좋고 수행시간은 느리고 반복은 상대적으로 가독성은 좋지 않고 수행시간은 빠르다는 특징을 알 수 있습니다.