

# 과제 #02

확장형 고성능 컴퓨팅 (2021년도 2학기, M1522.006700, M3239.002300)

Due: 2021년 10월 13일(수) 23시 59분

## 1 Representations of Numbers (40점)

int, long, float, double, long double 자료형에 해당하는 수를 입력받아, 이진수 표현으로 출력하는 프로그램을 작성하라. 디렉토리 numbers 안에 뼈대코드 int.c, long.c, float.c, double.c, longdouble.c가 주어진다. long double은 x86 80-bit extended precision을 따른다고 가정한다. 다음은 실행 예시이다.

```
$ make
$ ./int
2020
00000000000000000000000011111100100
$ ./long
3239003800001
000000000000000000000000101111001000100011101010111100000111000001
$ ./long
-1
1111111111111111111111111111111111111111111111111111111111111111111111
$ ./float
-1.0
1011111110000000000000000000000000
$ ./float
-1.99999988079
101111111111111111111111111111111111
$ ./double
3.14
0100000000001001000111101011100001010001111010111000010100011111
$ ./longdouble
3.14
01000000000000001100100011110101110000101000111101011100001010001111010111000011
```

자료형이 표현가능한 범위를 초과하거나, inf나 NaN 따위의 예외적인 경우는 테스트하지 않는다.

## 2 Timer (40점)

능숙한 시간 측정은 고성능 컴퓨팅에 필수적인 요소이다. 체육시간에 사용하는 스톱워치와 비슷한 기능을 가진 timer.c를 완성하라. timer.c는 다음 함수들을 가진다.

- void timer\_init(int n)  
N개의 스톱워치를 생성하고 0초로 초기화한다. 스톱워치의 번호는 0부터 N - 1까지이다.
- void timer\_finalize()  
생성되어 있던 스톱워치를 모두 삭제한다.

- `void timer_start(int idx)`  
`idx`번째 스톱워치의 시간을 흐르게 한다. 이미 흐르고 있다면 아무 동작도 하지 않는다.
- `void timer_stop(int idx)`  
`idx`번째 스톱워치의 시간을 멈춘다. 이미 멈춰 있다면 아무 동작도 하지 않는다.
- `double timer_read(int idx)`  
`idx`번째 스톱워치의 시간을 `double`로 반환한다. 단위는 초(second)이다.
- `void timer_reset(int idx)`  
`idx`번째 스톱워치의 시간을 멈추고 0초로 초기화한다.

시간 측정에는 `clock_gettime` 함수를 사용하라. `clk_id` 인자는 `CLOCK_MONOTONIC`을 사용하라. 함수의 자세한 사용법은 `man clock_gettime`을 참고하면 된다.

`N`에 특별한 제한이 없음에 유의하라. 즉, `N`개의 스톱워치의 정보를 저장할 공간을 동적 할당(*e.g.*, `malloc`)해야 한다.

명시하지 않은 예외처리는 하지 않아도 된다. 예를 들어, `timer_init`을 2번 연속으로 호출하거나, `timer_init`을 호출하지 않고 다른 함수를 호출한다거나 하는 경우는 테스트하지 않는다.

`main.c`에는 `timer.c`를 테스트하는 코드가 구현되어 있다. `main`을 실행하여 통과되는 경우 만점, 그렇지 않은 경우 0점이 부여된다. 다음은 테스트를 통과하는 실행 예시이다.

```
$ make
cc -Wall -c -o timer.o timer.c
cc -Wall main.c timer.o -lm -o main
$ ./main
Test 1-1 SUCCESS (result=0.000000, expected=0.000000)
...
Test 6-2 SUCCESS (result=0.100139, expected=0.100000)
ALL SUCCESS!
```

### 3 Measuring Performance (10점)

- (5점) 실습 서버에 장착된 CPU는 무엇인가? 실습서버의 a00 노드를 기준으로 한다.
- (5점) FLOPS(Floating Point Operations Per Second)는 초당 수행한 floating-point 연산의 횟수를 말한다. 실습 서버의 CPU는 1 코어가 1 사이클에 2개의 floating-point 덧셈을 수행할 수 있다. 실습 서버의 1 코어에서 floating-point 덧셈만 수행하는 경우, 이론상 달성할 수 있는 최고 성능(Theoretical peak FLOPS)은 몇 FLOPS인가?  
(Hint: 벡터 연산을 사용하지 않는다고 가정한다. 또한, 1 코어만 사용하는 경우 Intel CPU는 Base Frequency가 아닌 Max Turbo Frequency로 동작한다.)

### 4 Submission Instruction

- Section 1의 코드 `int.c`, `long.c`, `float.c`, `double.c`, `longdouble.c`와 Section 2의 코드 `timer.c` 그리고 Section 3의 보고서 `report.pdf`를 한 파일로 압축하여 ETL에 제출한다.
- 첨부 파일명은 계정을름 `HW02.zip`으로 한다. (*e.g.*, `shpc99_HW02.zip`)
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있다.
- Grace day를 사용하려면 본인이 숙제를 제출한 날에 조교에게 메일(`shpc21@aces.snu.ac.kr`)로 알려야 한다. 메일 없이 제출한 경우 다음 숙제를 위해 아낀 것으로 판단, 미제출 처리된다.