

# 어셈블리어언어및실습

## 숙제**2**

Kon-Woo Kwon  
Department of Computer Engineering  
Hongik University

# 숙제2

---

- 다음 정수 연산을 수행하는 SPARC 어셈블리 프로그램을 작성하시오.

$$\text{result} = ((A \bmod B) + (C - D)) \times 12 \div (E + F)$$

‘mod’ 는 나머지 연산 (C 언어의 경우 % 연산자)

- 숙제 제출기한: 10월 16일 금요일 오후 11시 59분까지
- 숙제 제출 방법
  - ❖ submit konwoo asm\_02
  - ❖ 소스파일만 제출, 화면 캡처 필요없음

# 숙제2 – 조건1

- main 함수에 들어온 이후에 A, B, C, D, E, F 는 각각 %i0, %i1, %i2, %i3, %i4, %i5 레지스터에 있다고 가정.
- 연산결과 (result) 는 %i0 레지스터에 들어가야 함

```
.global main
main:  save  %sp, -96, %sp

... 계산 코드 작성...

ret
restore
```

← 이 위치에 도착하면 A-F 값들이 %i0-%i5 에 들어있다고 가정.

← 코드가 끝나기 전에(ret 명령어 이전) 계산 결과를 %i0에 넣어둘 것.

# 숙제2 – 조건2

---

- 사용 가능한 레지스터들
  - ❖ %g0-%g4
  - ❖ %l0-%l7
  - ❖ %i0-%i5 (A부터 F의 값이 들어 있음)
  - ❖ %o0-%o5 (함수 호출 위해서 사용. 함수 호출 이후에는 값이 변경됨)
- A~F는 모두 32bit signed integer
- 곱셈 연산의 결과는 하위 32bit만 사용하고 상위 32bit는 무시함.
- 나눗셈 연산 등 모든 연산의 결과는 정수로 생각할 것.
- SPARC V8 의 곱셈 명령어는 사용하지 말 것.

# 숙제2 – 힌트

- 코드를 완성 후 어셈블하여 실행하면 제대로 된 결과가 안나올 것임 (%i0-%i5 에 원하는 값이 들어있지 않으므로...)
- 다음과 같이 코드를 변형하여 테스트 해볼 수 있음.

```
.global main

main:  save  %sp, -96, %sp
      mov  14, %i0
      mov  2, %i1
      mov  400, %i2
      ...

      ... 계산 코드 작성...

result:
      ret
      restore
```

테스트에 써 볼 임의의 A~F의 값을 %i0-%i5에 넣음

gdb에서 이 위치에 breakpoint 걸고 %i0 의 값을 확인

- 숙제 제출시 A~F를 임의로 세팅하는 코드는 반드시 삭제하고 제출!

# 제출 기한 엄수

---

- 10월 16일 금요일 23시 59분까지 제출
  - ❖ `submit konwoo asm_02`
- 기한 내 제출못했을 경우, 다음과 같이 제출가능하나 감점
  - ❖ `submit konwoo asm_02_late`
    - 10월 17일 토요일 23시 59분까지 제출: (100점 기준) 30점 감점
    - 10월 18일 일요일 23시 59분까지 제출: (100점 기준) 60점 감점
    - 10월 19일 월요일 23시 59분까지 제출: (100점 기준) 90점 감점
    - 이외의 경우 : (100점 기준) 100점 감점