Kiott 사물인터넷 환경의 임베디드 개발자 과정

2018-02-27
4회차 과제

정유경

과제1. 배운 내용 복습

- Goto
- Pipeline
- For문

Goto 제어문

- 중간의 코드는 무시하고 원하는 부분으로 건너뛰어야 경우에 사용하는 제어문
- goto를 적절히 활용하면 중복되는 코드를 없앨 수 있고, 특히 에러 처리에 매우 유용하기 때문에 리눅스 커널에서 자주 사용함

Pipeline

중첩해서 사용해 보자!

- 메모리에 있는 코드를 읽어오고, 해석하고, 실행하는 것을 Fetch – Decode - Execute 라고 표현함
- Fetch – Decode - Execute의 3단계가 가장 기본이다.
- 이때 3개의 어셈블리 명령어가 수행되려면 총 9cycle이 필요하다.

  (각 단계별로 CPU clock을 한 cycle을 소모한다고 가정)

→ 파이프 라인을 구성하여 시스템을 동작시키면 3개의 명령어를 처리하는데 5 cycle만 필요하다. (성능향상)

- 32bit ARM 명령어라면 주소 증가는 4byte씩

3단계 파이프라인의 경우
총 10 cycle 동안 명령어 실행(Execute)은 8번!

Instruction	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	Cycle 6	Cycle 7	Cycle 8	Cycle 9	Cycle 10
ADD	Fetch ADD	Decode ADD	Execute ADD							
SUB		Fetch SUB	Decode SUB	Execute SUB						
AND			Fetch AND	Decode AND	Execute AND					
XOR				Fetch XOR	Decode XOR	Execute XOR				
MOV					Fetch MOV	Decode MOV	Execute MOV			
ADD						Fetch ADD	Decode ADD	Execute ADD		
SUB							Fetch SUB	Decode SUB	Execute SUB	
AND								Fetch AND	Decode AND	Execute AND

For 문

- for(초기화; 조건식; 증감식)으로 구성됨
- For(;;) → 무한루프

과제2. 프로그래밍 연습문제
(for 문으로 작성)

1. 스키장에서 스키 장비를 임대하는데 37500원이 든다 3일 이상 이용할 경우 20%를 할인 해준다. 일주일간 이용할 경우 임대 요금은 얼마일까?

```c
#include <stdio.h>

int TOTAL_F(int);

int main(void)
{
    int result = 0, day = 0;
    int i = 0;
    printf("최대 몇일간 이용하실 계획인가요: ");
    scanf("%d", &day);
    result = TOTAL_F(day);
    for(i=1;i<day+1;i++)
    printf("%d 일간 이용할 경우 임대요금은 %d원 입니다.\n", i, result);
    return 0;
}

int TOTAL_F(int day)
{
    double total = 37500 * day;
    if (day >= 3)
        total *= 0.8;
    return total;
}
```

3. 1 ~ 1000사이에 3의 배수의 합을 구하시오

```c
#include <stdio.h>

int SUM_F(int, int);

int main(void)
{
    int i = 0, sum = 0;

    for (i = 1; i < 1001; i++)
        sum = SUM_F(i, sum);

    printf("\n1~1000사이의 3의 배수의 합은 %d 이다.\n\n", sum);
    return 0;
}

int SUM_F(int i, int result)
{
    if (i % 3 == 0)
        result += i;
    return result;
}
```

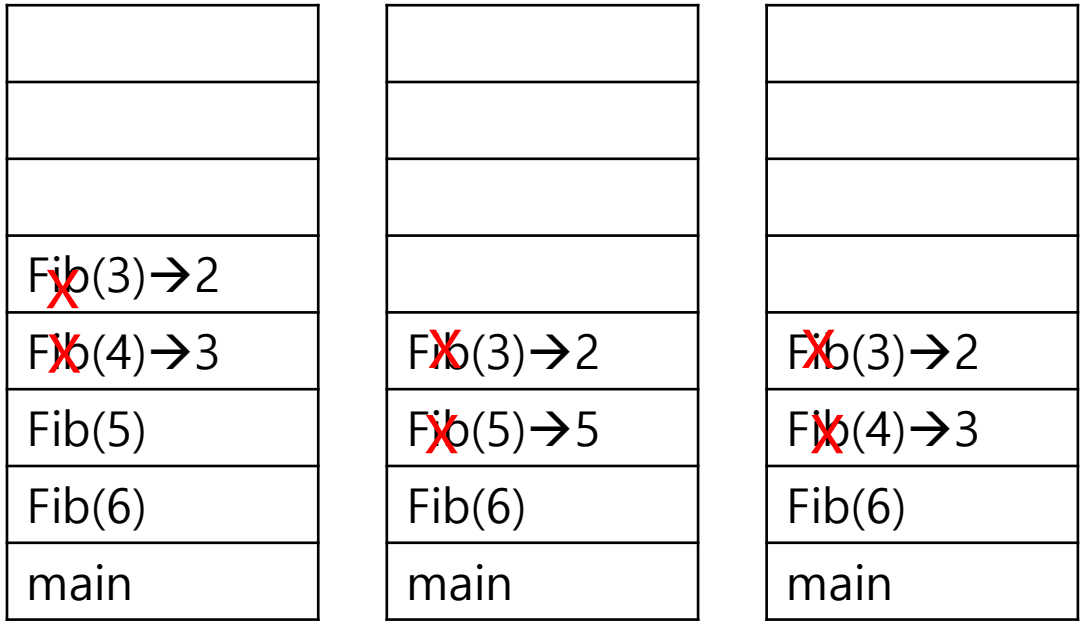
4. 1 ~ 1000사이에 4나 6으로 나눠도 나머지가 1인 수의 합을 출력하라.

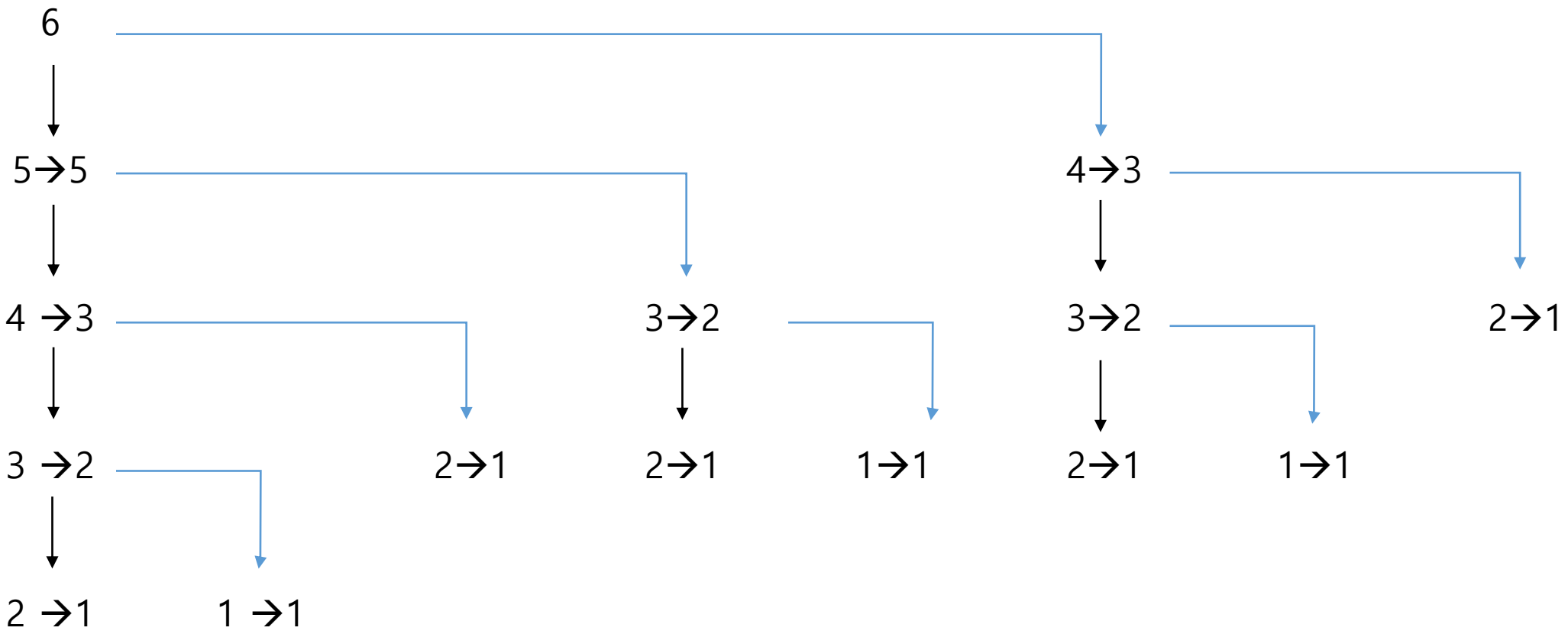
```c
#include <stdio.h>

int FUNC(int, int);

int main(void)
{
    int i = 0, sum = 0;

    for (i = 1; i < 1001; i++)
        sum = FUNC(i, sum);

    printf("\n1~1000사이에서 4나 6으로 나누어도 나머지가 1인 수의 합은 %d이다.\n", sum);

    return 0;
}

int FUNC(int i, int result)
{
    if (i % 4 == 0 && i % 6 == 0)
    {
        i++;
        result += i;
    }
    return result;
}
```

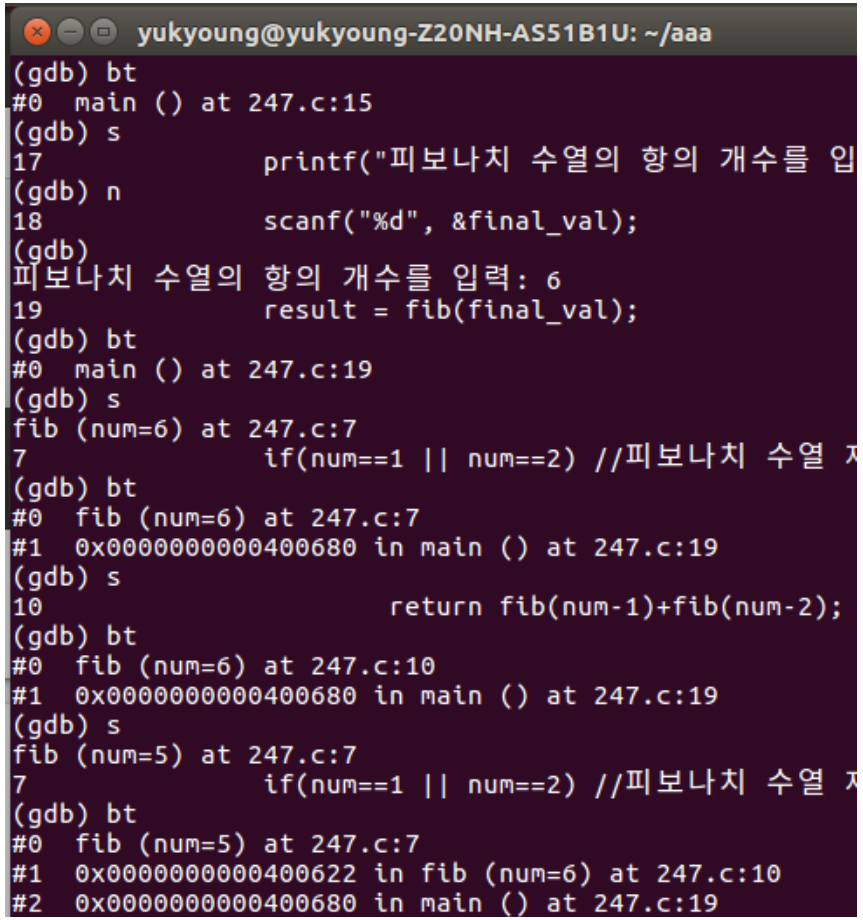
10. 구구단을 만들어 보시오

```c
#include <stdio.h>
void GUGU();

int main(void)
{
    printf("구구단을 출력합니다.\n\n");
    GUGU();
    return 0;
}

void GUGU()
{
    int i = 0, j = 0;

    for (i = 1; i < 10; i++)
    {
        printf("\n\n구구단 %d단입니다.\n\n", i);
        for (j = 1; j < 10; j++)
            printf("%d x %d = %d\n", i, j, j*i);
    }
}
```

과제3. fib 함수 동작 분석
(디버깅 및 그림 그리기)




6
5→5
4→3
4 →3
3→2
3→2
2→1
3 →2
2→1
2→1
1→1
2→1
1→1
2 →1
1 →1



← 현재 stack에 메인이 있다

← N명령어를 사용하여 함수 내부로 들어가지 않는다

← 항의 개수 6개를 입력받았다.

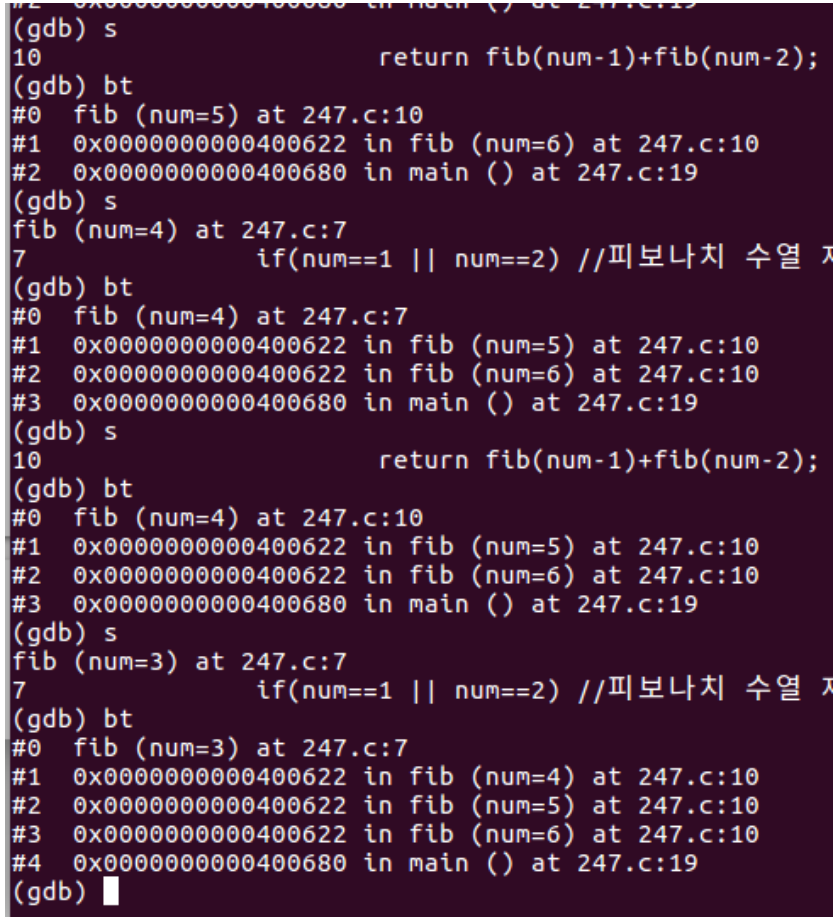
← bt 명령어로 스택의 현재상태를 본다

← s명령어로 함수 내부로 진행한다

← Fib함수에 6가 걸려있다

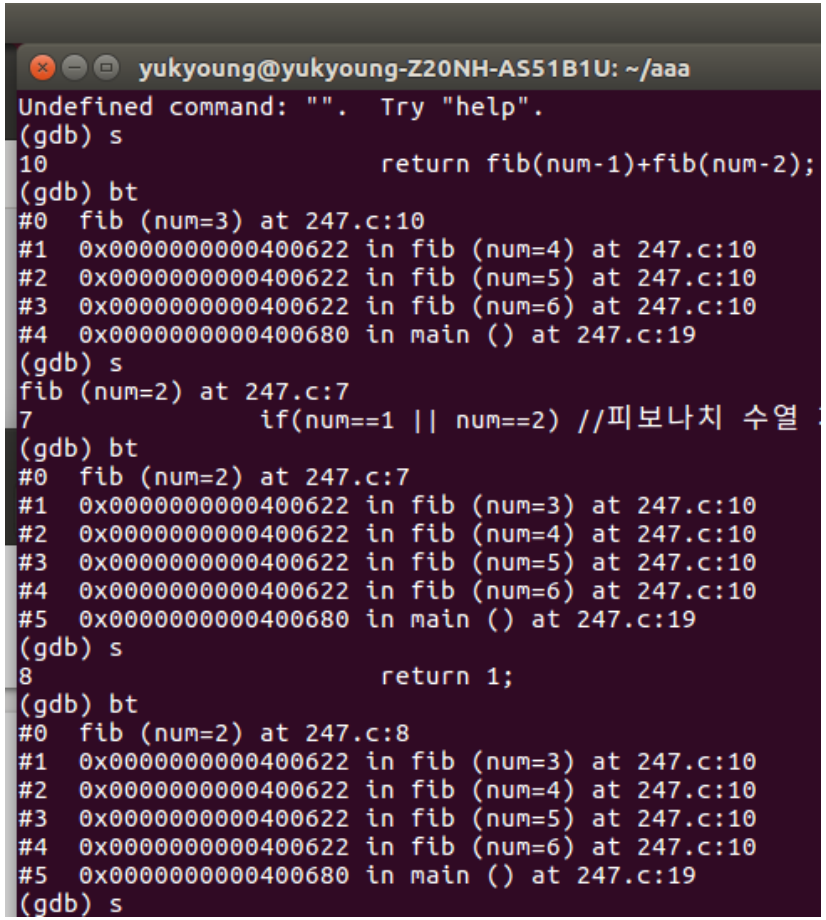
←num-1을 진행하여 Fib num=5가 호출된다

← 확인해보면, 메인에 fib6에 fib5가 쌓여있다



←Num-1을 진행하여 Fib num=4가 호출된다

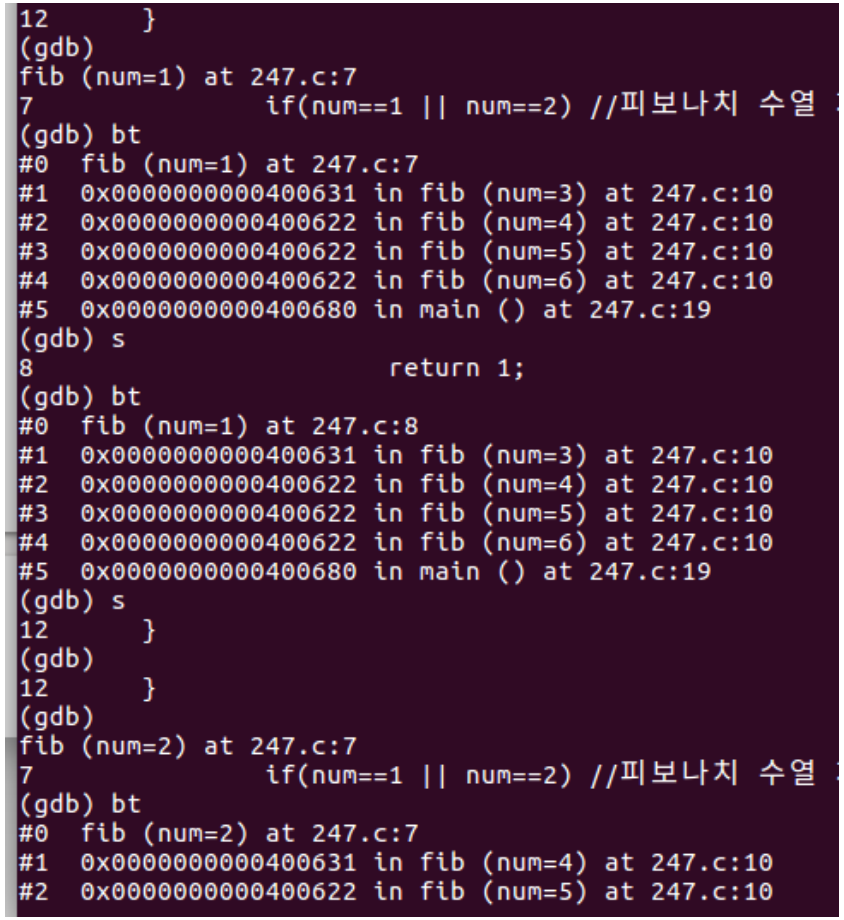
← num-1을 진행하여 Fib num=3이 호출된다



← Num-1되어 2이므로 if에 걸려서 1을 반환한다

← Stack에 차례로

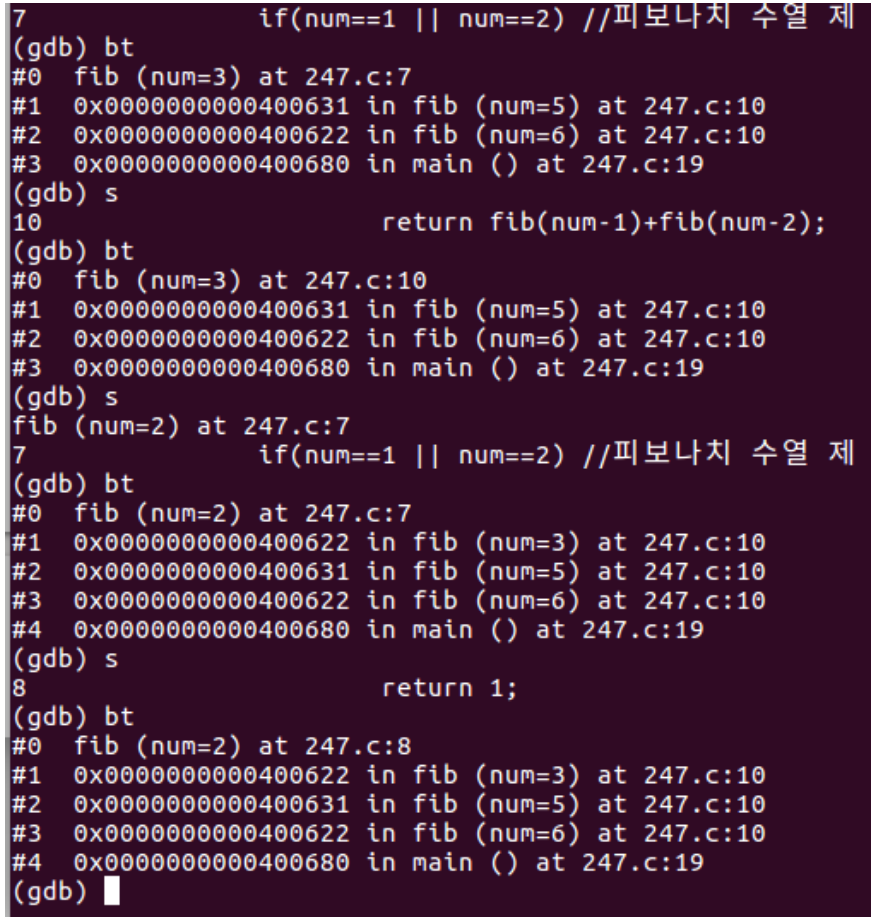
메인, fib6, fib5, fib4, fib3,fib2가 쌓여있음을 알수있다



←3에서 n-2 진행하여 fib num=1이 호출되었다

←If에 걸리므로 1을 반환한다

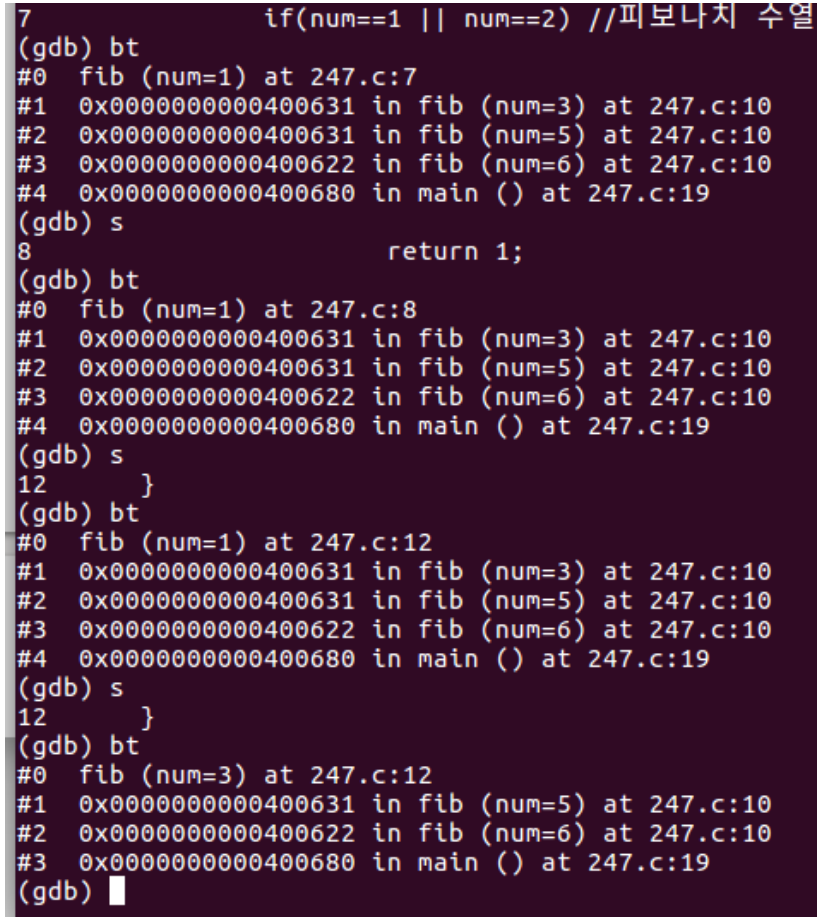
←4에서 n-2진행하여 fib num=2가 호출되었다
If에 걸리게 되어 1을 반환할 것이다



←5에서 n-2진행하여 fib num=3이 호출되었다

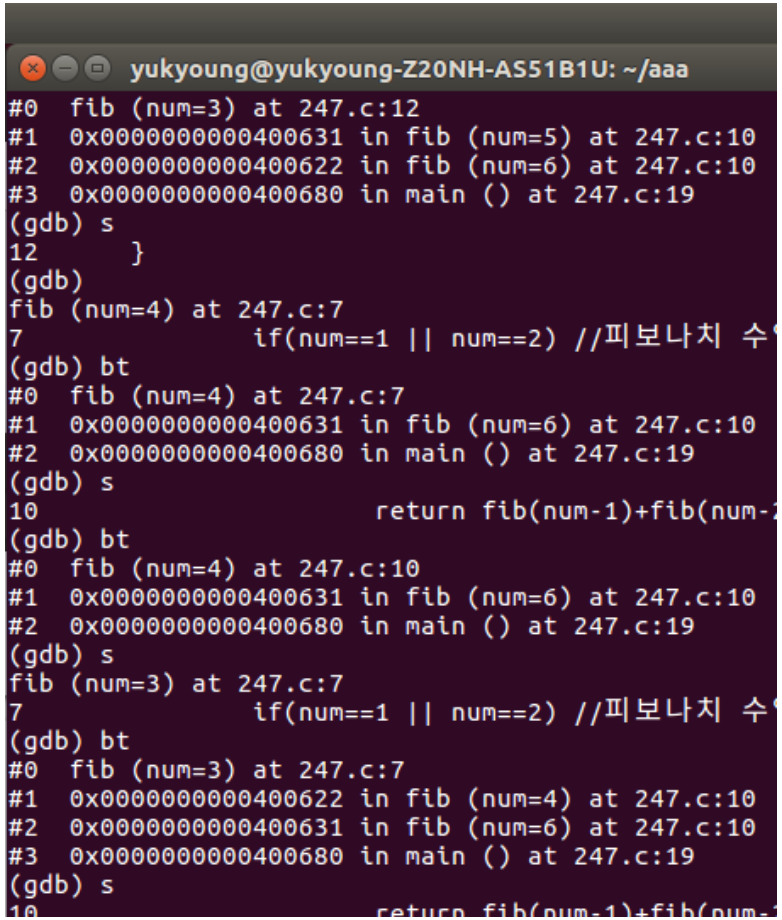
←3에서 n-1진행하여 fib num=2 호출되었다

←If에 걸리므로 1 반환한다



←3에서 n-1진행하여 fib num=1호출되었다

←If에 걸려서 1을 반환한다



←fib num3과 fib num5의 값을 합한다

←6에서 n-2진행하여 fib num=4가 호출되었다

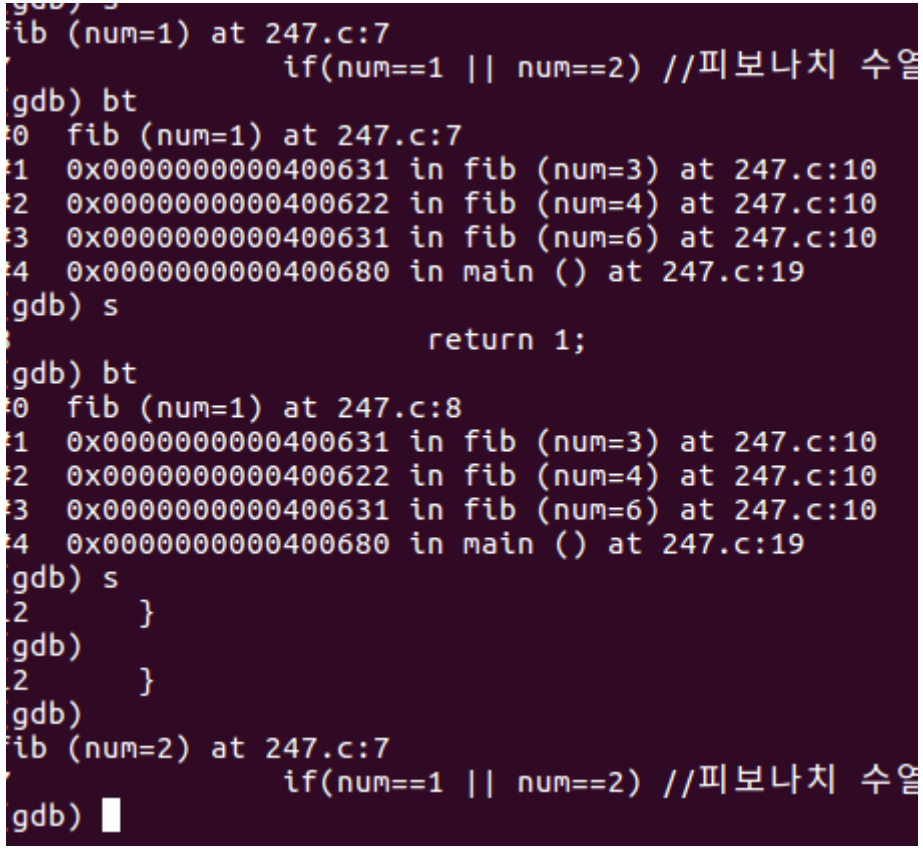
←현재 stack에는 메인과 fib6이 걸려있다

←4에서 n-1진행하여 fib num=3이 호출되었다

←현재 stack에는 메인, fib6, fib4가 쌓여있다

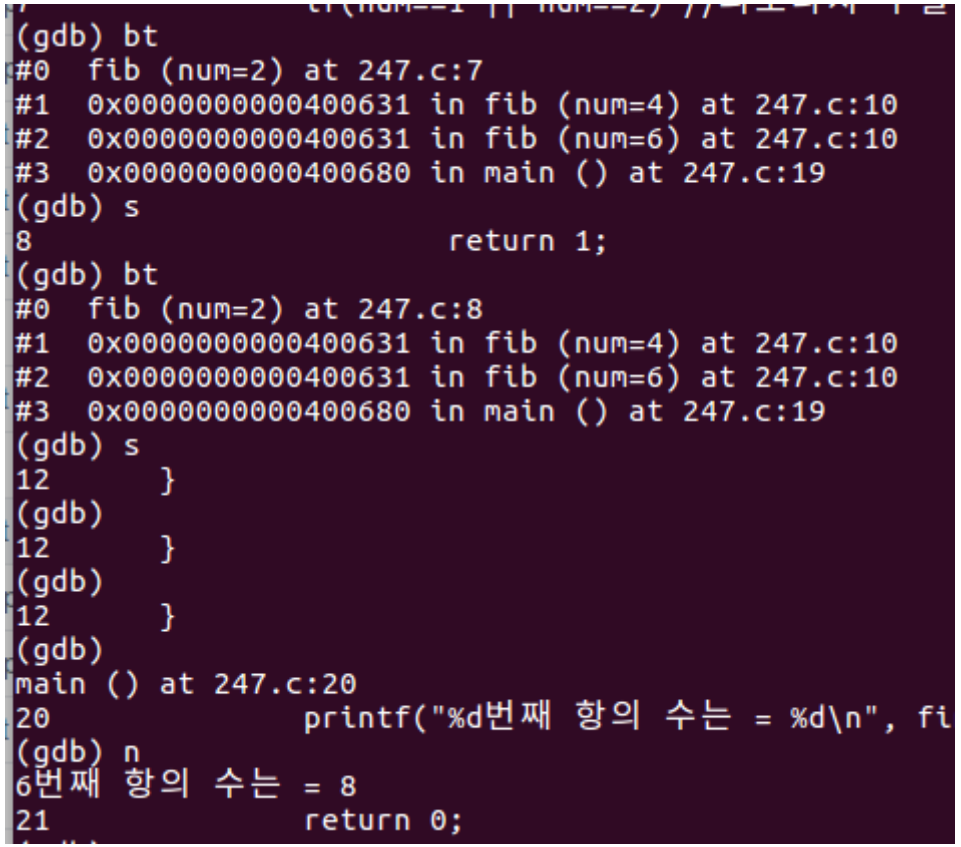
```
(gdb) bt
#0  fib (num=3) at 247.c:10
#1  0x0000000000400622 in fib (num=4) at 247.c:10
#2  0x0000000000400631 in fib (num=6) at 247.c:10
#3  0x0000000000400680 in main () at 247.c:19
(gdb) s
fib (num=2) at 247.c:7
7               if(num==1 || num==2) //피보나치 수
(gdb) bt
#0  fib (num=2) at 247.c:7
#1  0x0000000000400622 in fib (num=3) at 247.c:10
#2  0x0000000000400622 in fib (num=4) at 247.c:10
#3  0x0000000000400631 in fib (num=6) at 247.c:10
#4  0x0000000000400680 in main () at 247.c:19
(gdb) s
8                       return 1;
(gdb) bt
#0  fib (num=2) at 247.c:8
#1  0x0000000000400622 in fib (num=3) at 247.c:10
#2  0x0000000000400622 in fib (num=4) at 247.c:10
#3  0x0000000000400631 in fib (num=6) at 247.c:10
#4  0x0000000000400680 in main () at 247.c:19
(gdb) s
12      }
(gdb) bt
#0  fib (num=2) at 247.c:12
#1  0x0000000000400622 in fib (num=3) at 247.c:10
#2  0x0000000000400622 in fib (num=4) at 247.c:10
#3  0x0000000000400631 in fib (num=6) at 247.c:10
#4  0x0000000000400680 in main () at 247.c:19
(gdb)
```

← 3에서 n-1진행하여 fib num=2가 호출되었다
If에 걸리므로 1을 반환한다



← 3에서 n-2진행하여 fib num=1이 호출되었다
If에 걸리므로 1을 반환한다

←현재 stack에는
메인, fib6, fib4, fib3 fib1이 쌓여있따



← fib2 에서 1, fib1에서1 총 2가 fib3에 전달된다

← 이를 fib4에 더하고 fib6에 더한다

← 최종 결과인 8이 출력된다

피보나치 수열의 6번째 항은 8이다