Xilinx Zynq FPGA, TI DSP, MCU기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - Innov (이상훈)
gcccompil3r@gmail.com
학생 - 이유성
dbtjd1102@naver.com

swich문
전등 스위치 같은 것이라고 생각하면 된다.
a를 누르면 a가 들어오고
b를 누르면 b가 들어온다고 보면 되겠다

break문은 현재 switch문을
그냥 빠져나가겠다는 것을 의미한다

```
#include<stdio.h>
int main(void){

	int num;

	printf("상담은 1, 결제는 2, 교체는 3 : ");
	scanf("%d",&num);

	switch(num)
	{
		case 1:
		printf("상담 연결 중 입니다\n");
		break;
		case 2:
		printf("결제 중 입니다\n");
		break;
		case 3:
		printf("교체 연결 중 입니다\n");
		break;

		default:
		printf("잘못 누르셨습니다\n");
		}
		return 0;
	}
```

static문

```
#include <stdio.h>
void count_static_value(void)
{
	static int count = 1;
	printf("count = %d\n", count);
	count++;
}
int main(void)
{
```

```
	int i;
	for(i = 0; i < 7; i++)
		count_static_value();
}
```

전역 변수란 무엇인가 ?
변수의 위치에 관계 없이 data값을 참조할 수 있음
Thread Programming시 문제 발생
동기화 메커니즘이 필요한 이유 ?!
Memory의 data 영역에 load됨

```
#include<stdio.h>
	int number  = 6;

void func(){

	printf("%d\n",number+1);



	}

int main(){

	printf("%d\n",number);

	func();

	return 0;
	}
```

static 키워드의 역할은 ?
정적 변수
static 변수는 전역 변수와 마찬가지로 data 영역에 load됨
지역 변수를 static으로 선언했다면
이 변수를 선언한 함수내에서만 접근 가능함

```
#include <stdio.h>
void count_static_value(void)
{
	static int count = 1;
	printf("count = %d\n", count);
	count++;
```

```
}
int main(void)
{
	int i;
	for(i = 0; i < 7; i++)
		count_static_value();
}
```

while을 통한 무한 Loop는 어떻게 ?
조건을 무조건 참으로 만들면 된다
true는 1
false는 0

break로 조건을 줘서 멈출 수 있다.

```
#include<stdio.h>
int main(void)
{

	int number =0;
	while(1)
	{
		printf("%d\n",number);
		number++;
		if(number  ==100){
		break;
		}

	}
	return 0;
}
```

continue문은 왜 필요할까 ?
해당 case는 제끼고 반복은 계속하고 싶어서

```
#include<stdio.h>
int main(void)
{
	int number =0;
	while(1)
	{
```

```
                if(number ==5)
                continue;
                printf("%d\n",number);
                number++;
                if(number ==10)
                break;

        }
        return 0;
}
```

여기서
number가 5인 경우
아래를 수행하지 않고
while의 조건을 검사하러 돌아감

while문은 초기화, 조건식, 증감식이 보기 불편함
for(초기화; 조건식; 증감식)으로 구성됨
간결성의 미학 for(초기화; 조건식; 증감식)으로 구성됨

for(; ;) 조건을 막론하고 for문의 조건부가 비어 있으면 언제든지 무한 Loop이 됨

goto는 말이 많은 녀석이다
주로 Kernel에서 많이 사용하며 Buffer를 사용하는 곳에서 주로 사용함

```
#include<stdio.h>
int main(void){
int i,j ,number =0;
for(i=1;i<4;i++){
 for(j=1;j<4;j++){
  if(i==2)
  goto error_handler;
  printf("i*j = %d",i*j);

 }
}
error_handler:
 printf("Error를 처리중\n");

return 0;
}
```

재귀 함수 호출이란 ?
사용한 함수를 다시 호출하는 방식
Program 구현상 반드시 필요한 경우가 있음
무조건 좋지는 않지만, 편의가 따름
성능등을 잘 고려해서 구현해야함

피보나치수열

```
#include int fib(int num){
if(num == 1 || num == 2)
return 1;
 else
 return fib(num – 1) + fib(num – 2);
}
int main(void){
int result, final_val;
 printf(“피보나치 수열의 항의 개수를 입력하시오 : “);
 scanf(“%d”, &final_val);
result = fib(final_val);
printf(“%d번째 항의 수는 = %d\n”, final_val, result);
 return 0;
}
```