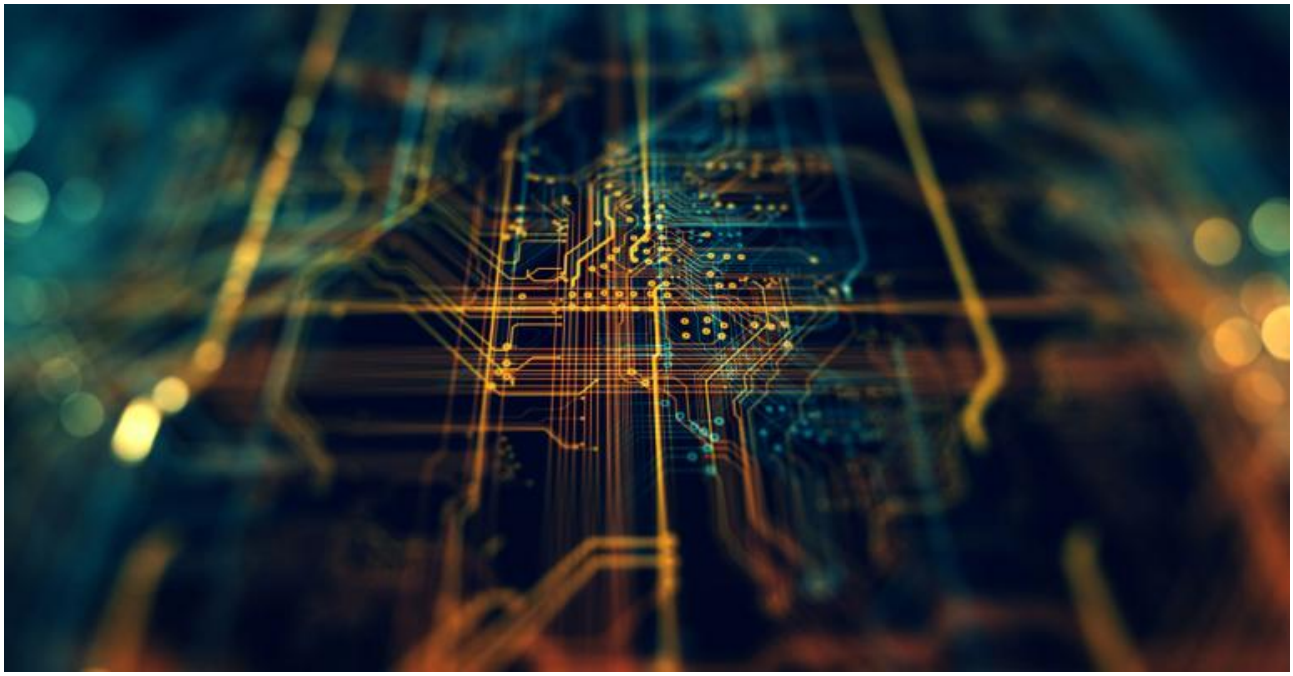
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정



2018.03.28
25 일차
강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – 신민철
akrn33@naver.com

adv_tech4.c

```
#include<stdio.h>

#include<stdlib.h>

typedef struct{
	int score;
	char name[20];
}ST;

typedef struct
{
	int count;
	char name[20];
	int score[0];
}FLEX;

int main(void)
{
	int i;
	FLEX *p = (FLEX*) malloc(4096);
	printf("%d",sizeof(FLEX));
	for(i = 0; i < 100000; i++)
	{
		p->score[i] = i+1;
		printf("%d\n",p->score[i]);
	}
	return 0;

}
/*
	이코드는 동적할당을 크게잡아서 동적할당수를 줄이고 그시간을 없애기 위해서
	구조체의 끝이자 새로운시작 스코어 0
```

```
*/

sem.c
#include"sem.h"

int main(void)
{
	int sid;
	sid = CreateSEM(0x777);//세마포어아이디에 권한을 줌공엑스 77
은 권한

	printf("before\n");

	p(sid);
	printf("Enter Critical Section\n");

	getchar();
	v(sid);

	printf("after\n");
	return 0;
}
/*
	컴파일 방법 gcc semlib.c sem.c
*/

sem.h
#include<sys/types.h>
#include<sys/ipc.h>
#include<sys/sem.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<errno.h>
```

```
#define SEMPERM 0777

int CreateSEM(key_t semkey);
int p(int semid);
int v(int semid);
```

semlib.c

```
#include "sem.h"

int CreateSEM(key_t semkey)
{
	int status = 0,semid;
	if(semid = (semget(semkey, 1, SEMPERM | IPC_CREAT | IPC_EXCL)) == -1){//해당키값으로 세마포어가 있으면 씹어라
		if(errno == EEXIST)//존재한다면
			semid = semget(semkey, 1, 0);//세마포어 존재하는것을 가져오겠다
	}
	else
		status = semctl(semid, 0, SETVAL, 2);//셈씨티엘을 하게된다면 현재 세마포어 아이디를 가운데에 있는걸로 세팅해준다는거
	if(semid == -1 ||status == -1)
		return -1;
	return semid;//키값 777 을가지고 아이디를 얻은다음에 아이디값을 리턴함
}

int p(int semid)
{
	printf("%d",3);
```

```
	struct sembuf p_buf = {0, -1, SEM_UNDO};//종료할 때 건드리는값이 존재한다면 0 으로 초기화해주라고
	if(semop(semid, &p_buf, 1) == -1)//세마포어 값을 1 증가시키라는거//덧셈패턴
		return -1;//실패하면 -1 을 리턴
	return 0;
}

int v(int semid)
{
	struct sembuf p_buf = {0, 1, SEM_UNDO};//뺄셈할때 패턴
	if(semop(semid, &p_buf, 1) == -1)
		return -1;
	return 0;
}
```

recv.c

```
#include "shm.h"

int main(void)
{
	int mid;
	SHM_t *p;

	mid = CreateSHM(0x888);

	p = GetPtrSHM(mid);

	getchar();
	printf("이름 : [%s], 점수 : [%d]\n", p->name,p->score);

	FreePtrSHM(p);
	return 0;
```

```
}
/*
	컴파일 방법
	gcc -o send shmlib.c send.c
	gcc -o recv shmlib.c recv.c
	터미널 두 개 띄우고 ./send 이후에 ./recv
*/
```

send.c

```
#include "shm.h"

int main(void)
{
	int mid;//메모리아이디 Memory I D mid
	SHM_t *p;//SHM_t 의 포인터형.

	mid = OpenSHM(0x888);//페이지프레임에 접근할 아이디값을

	p = GetPtrSHM(mid);//p 가 가리키는 곳의 주소에

	getchar();
	strcpy(p->name,"아무개");
	p->score = 93;

	FreePtrSHM(p);

	return 0;
}
```

shm.h

```
#include<sys/ipc.h>
#include<sys/shm.h>
```

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<errno.h>

typedef struct
{
	char name[20];
	int score;
}SHM_t;

int CreateSHM(long key);
int OpenSHM(long key);
SHM_t* GetPtrSHM(int shmid);
int FreePtrSHM(SHM_t* shmptr);
```

shmlib.c

```
#include "shm.h"

int CreateSHM(long key)
{
	return shmget(key, sizeof(SHM_t), IPC_CREAT|0777);
}
int OpenSHM(long key)
{
	return shmget(key, sizeof(SHM_t), 0);//OS 로부터 공유메모리를 받아오는 함수. 공유메모리의 ID 값을 반환함. 공유메모리를 받아오기 위해서 key, size, flag 인자를 지정함. key 는 int 값으로 돌려받을 공유메모리의 번호를 지정하는 값, size 는 공유메모리의 크기를 지정하는 값이고, flag 에는 여러 옵션값들이 들어갈수있음 IPC_CREAT 은 공유메모리가 없으면 만들라는거,IPC_EXCL 을 사용하면 키가있으면 메모리를 만들지 말라는 옵션.
}
```

```
SHM_t* GetPtrSHM(int shmid)
{
	return (SHM_t*)shmat(shmid,(char*)0,0);//
```
shmat(id,*addr,flags) addr 인자는 공유메모리가 프로세스 어디에 위치하여야 하는지 나타내고, null 값으로 받게 된다면 OS 가 그 위치를 지정하게 된다. SHM_RDONLY 를 사용하면 공유메모리를 읽기 전용으로 첨부하게 된다.
```
}
int FreePtrSHM(SHM_t* shmptr)
{
	return shmdt((char*)shmptr);
}
```