2018. 3. 27 화 – 24 회차

과정 : TI, DSP, Xilinx Znq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈
gcccompil3r@gmail.com
Stu. 정상용
fstopdg@gmail.com

Linux 운영체제

1. signal

→ signal(SIGINT, my_sig)
   signal : manual for some situation.
   Above case : If SIGINT occurs, do my_sig. (SIGINT : Ctrl + v)
→ First return for signal is NULL (Check out ‘ex’)
    Second one is my_sig.

Q.
1. what does ‘pause()’ do?
2. what does ‘for(;;) do in this program?

Ex.

```
#include <signal.h> //it’s for ‘signal’
#include <stdio.h>
#include <unistd.h>

void my_sig(int signo)
{
    printf("my_sig called\n");
}

void my_sig2(int signo)
{
    printf("my_sig2 called\n");
}

int main(void)
{
    void(*old_p)(int);
    void(*old_p2)(int);
    old_p = signal(SIGINT, my_sig);
    pause();
    old_p2 = signal(SIGINT, my_sig2);
    pause();
    old_p2 = signal(SIGINT, old_p); // If ‘old_p’ changes to ‘old_p2’, my_sig2 do.
    pause;
    for(;;)
        pause();
    return 0;
}
```

2. kill -9 PID

→ 'GOD' : He could kill anyone who he wanna kill.
→ how to check : ./a.out → open terminal → ps -ef | grep a.out(check PID) → kill -9 PID(he kill the program) → ps -ef | grep a.out(check the program was killed again)

Ex.

```
#include <signal.h>
#include <stdio.h>
#include <unistd.h>

int main(void)
{
    signal(SIGINT, SIG_IGN);
    pause();
    return 0;
}
```

Q
1. what is 'kill -2'?
2. what is 'SIG_IGN'?

3. review : goto

→ it works by 'goto' & 'label'
→ perror : show me the reason why system call works.
Ex.

```
#include <signal.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

int main(void)
{
    int ret;
    char buf[1024];
    if((ret = read(0, buf, sizeof(buf))) > 0)
        goto err;
    return 0;

    err:
        perror("read()");  //()안에 들어있는 시스템콜에 대한 동작의 이유를 보여줌
        exit(-1);
}
```

4. the weakness of 'goto'

→ goto can not move out of function.
Because can not dismantle stack.

Ex.

```
#include <signal.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

void func(int fd, char *buf, int size)
{
   if((read(fd, buf, size) > 0))
      goto err;

   return ;
}



int main(void)
{
   int ret;
   char buf[1024];
   ret = func(0, buf, sizeof(buf));

   return 0;

   err:
       perror("read()");
       exit(-1);
}
```

5. setjmp(env) & longjmp (header file : <setjmp.h>

→ Upgrade version of goto
→ goto : longjmp,   label : env
→ First return value of setjmp is ‘0’
→ longjmp(env, 1) : setjmp(env) = 1

Ex.

```
#include <stdlib.h>
#include <setjmp.h>
#include <stdio.h>

jmp_buf env; //

void test(void)
{
   longjmp(env, 1);
}

int main(void)
{
   int ret;
```

```
    if((ret = setjmp(env)) == 0)
    {
       printf("this\n");
       test();
    }
    else if(ret > 0)
       printf("error\n");
    return 0;
}
```

6. setjmp(env) & longjmp : advanced

Q.

1. longjmp(env1, 2) → longjmp(env1, 0) : how to work?? and why??

Ex.

```
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <setjmp.h>

jmp_buf env1;
jmp_buf env2;

void test1(void)
{
        longjmp(env1, 1);
}

void test2(void)
{
        longjmp(env1, 2);
}

void test3(void)
{
        longjmp(env2, 1);
}

int main(void)
{
        int ret;
        if((ret = setjmp(env1)) == 0)
        {
                printf("this\n");
                test1();
        }
        else if(ret == 1)
        {
                printf("1\n");
                test2();
        }
        else if(ret == 2)
```

```
			printf("2\n");
		else
		{
			printf("goto letsgo label\n");
			goto letsgo;
		}

		if((ret = setjmp(env2)) == 0)
		{
			printf("second label\n");
			test3();
		}
		else
			longjmp(env1, 3);
letsgo:
			goto err;

		return 0;

err:
		printf("Error!!!\n");
		exit(-1);
}
```

7. setjmp(env) & longjmp : advanced

below ‘longjmp’ in same function, it does not work. → it does not have any meaning.

Ex.

```
#include <fcntl.h>
#include <stdlib.h>
#include <setjmp.h>
#include <stdio.h>

jmp_buf env;

void test(void)
{
   int flag = -1;
   if(flag < 0)
      longjmp(env, 1);
   printf("call test\n");
}

int main(void)
{
   int ret;
   if((ret = setjmp(env)) == 0)
      test();
   else if(ret > 0)
      printf("error\n");
   return 0;
}
```

8. SIGALRM

→ if we type something within 3sec, it finishes.
  Or not, it prints.

Q.

1. What does ‘alarm(0)’ mean exactly??

Ex.

```
#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

void my_sig(int signo)
{
   printf("You must inset coin\n");
   exit(0);
}

int main(void)
{
   char buf[1024];
   int ret;
   signal(SIGALRM, my_sig);
   alarm(3);
   read(0, buf, sizeof(buf));
   alarm(0);
   return 0;
}
```

9. Game by using SIGALRM

Ex. what I made

```
#include <stdio.h>
#include <signal.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <time.h>
#include <stdlib.h>

void my_sig(int signo)
{
        printf("Game is done!!!\n");
        exit(0);
}

int main(void)
{
        char buf[1024];
        char buf_2[1024] = {0};
```

```
    int nread, ret, num, num2;

    srand(time(NULL));
    num = rand()%1000 + 1;
    //printf("num = %d\n", num);

    while(1)
    {
        signal(SIGALRM, my_sig);
        alarm(2);

        nread = read(0, buf_2, sizeof(buf_2));
        num2 = atoi(buf_2);

        if(num < num2)
            printf("down\n");

        else if(num > num2)
            printf("up\n");

        else if(num = num2)
        {
            printf("Felicidades!!!\n");
            break;
        }

    }

    return 0;
}
```

Ex2. What professor made

```
#include <time.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <signal.h>
#include <stdlib.h>
#include <stdbool.h>

void sig_handler(int signo)
{
    printf("You lose! Input should be within 1 second!\n");
    exit(0);
}

void make_game(int *data)
{
    *data = rand() % 100 + 1;
}

bool check_correct(int data, int cmp)
```

```
{
    if(data == cmp)
        return true;
    else
        return false;
}

void start_game(int data)
{
    char buf[32] = {0};
    bool fin;
    int i, cmp;

    for(i = 0; i < 10; i++)
    {
        signal(SIGALRM, sig_handler);
        printf("숫자를 맞춰봐!\n");
        alarm(1);
        read(0, buf, sizeof(buf));
        alarm(0);
        cmp = atoi(buf);

        fin = check_correct(data, cmp);

        if(fin)
        {
            printf("You Win!!!\n");
            exit(0);
        }
        else
        {
            if(data > cmp)
                printf("%d 보다 크다\n", cmp);
            else
                printf("%d 보다 작다\n", cmp);
        }
    }

    printf("You Lose!!! You Babo\n");
}

int main(void)
{
    int data;

    srand(time(NULL));
    make_game(&data);

    start_game(data);

    return 0;
}
```