

< 운영체제 과제 4 >

2017253041\_홍성우

1.

(1)

```
#include <stdio.h>
#include <sys/shm.h>
#include <sys/stat.h>

int main()
{
    int pid;
    int segment_id;
    char* shared_memory;
    const int segment_size = 4096;

    segment_id = shmget(IPC_PRIVATE,segment_size,S_IRUSR | S_IWUSR);
    printf("create shared memory : segment_id = %d\n", segment_id);
    shared_memory = (char*) shmat(segment_id,NULL,0);

    pid = fork();
    if(pid < 0)
        return -1;
    else if(pid == 0) {
        sprintf(shared_memory, "Hello Parent");
        if(shmdt(shared_memory) == -1) {
            fprintf(stderr, "Unable to detach\n");
        }
    } else {
        wait(NULL);
        printf("%s\n",shared_memory);
        if(shmdt(shared_memory) == -1) {
            fprintf(stderr, "Unable to detach\n");
        }
    }
    return 0;
}
```

```

u17041@solgae:~/prog$ ls -l
total 24
-rw-r--r-- 1 u17041 u_student 683 4월 16 22:05 1-1.c
-rwxr-xr-x 1 u17041 u_student 17024 4월 16 22:07 a.out
u17041@solgae:~/prog$ ./a.out
create shared memory : segment_id = 32826
*Hello Parent*
u17041@solgae:~/prog$ ls
1-1.c a.out
u17041@solgae:~/prog$

```

(2-1)

```

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MSGSZ 128

typedef struct msgbuf
{
    long mtype;
    char mtext[MSGSZ];
} message_buf;

int main(void)
{
    int msqid;
    int msgflg = IPC_CREAT | 0666;
    key_t key;
    message_buf sbuf;
    size_t buf_length;

    key = 1234;
    fprintf(stderr, "\nmsgget: Calling msgget(%#lx,%#o)\n", key, msgflg);
    if ((msqid = msgget(key, msgflg)) < 0) {
        perror("msgget");
        exit(1);
    } else {
        fprintf(stderr, "msgget: msgget succeeded: msqid = %d\n", msqid);
        sbuf.mtype = 1;
        strcpy(sbuf.mtext, "Did you get this?");
        buf_length = strlen(sbuf.mtext) + 1;
        if (msgsnd(msqid, &sbuf, buf_length, IPC_NOWAIT) < 0) {
            printf("%d, %d, %s, %d\n", msqid, sbuf.mtype, sbuf.mtext, buf_length);
            perror("msgsnd");
            exit(1);
        } else {
            printf("Message: \"%s\" Sent\n", sbuf.mtext);
            exit(0);
        }
    }
}

```

~  
~  
~  
~

"1-2-1.c" 42L, 894C 42,0-1 모두

```

u17041@solgae:~/prog$ ls -l
total 28
-rw-r--r-- 1 u17041 u_student 683 4월 16 22:05 1-1.c
-rw-r--r-- 1 u17041 u_student 894 4월 16 22:22 1-2-1.c
-rwxr-xr-x 1 u17041 u_student 16984 4월 16 22:22 a.out
u17041@solgae:~/prog$ ./a.out

msgget: Calling msgget(0x4d2,01666)
msgget: msgget succeeded: msqid = 7
Message: "Did you get this?" Sent
u17041@solgae:~/prog$ ls
1-1.c 1-2-1.c a.out
u17041@solgae:~/prog$

```

(2-2)

```

#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#include <stdlib.h>

#define MSGSZ 128

typedef struct msgbuf {
    long mtype;
    char mtext[MSGSZ];
} message_buf;

int main (void) {
    int msqid;
    key_t key;
    message_buf rbuf;

    key = 1234;
    if ((msqid = msgget (key, 0666)) < 0) {
        perror ("msgget");
        exit (1);
    }

    if (msgrcv (msqid, &rbuf, MSGSZ, 1, 0) < 0) {
        perror ("msgrcv");
        exit (1);
    }

    printf ("Received Message: %s\n", rbuf.mtext);
    exit (0);
}

```

```
u17041@solgae:~/prog$ ./a.out
Received Message: Did you get this?
u17041@solgae:~/prog$ ls
1-1.c 1-2-1.c 1-2-2.c a.out
u17041@solgae:~/prog$
```

(3)

shared memory는 계속해서 공유되는 메모리를 새로 지정한다. 하지만 message queue는 메시지를 지정된 장소에 저장, 인출하여 전달하고 다시 받는다. shared memory는 공유 영역에 있는 버퍼를 사용하여 메모리를 공유한다. 협력 프로세스들의 전형적인 예시이고 공유 메모리가 지정되면 일상적인 메모리와 같이 접근된다. message Queue는 서로 공유하지 않고 통신을 하며 동작을 동기화하는 기능을 제공한다.

2.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/wait.h>
#define MAX_LINE 80
int main(void)
{
    char input[MAX_LINE];
    char* args[MAX_LINE/2 + 1];
    char* token;
    int i;

    pid_t pid;
    while(1) {
        if(pid != 0) {
            printf("mysh > ");
            if(fgets(input, sizeof(input)-1, stdin) == NULL) {
                exit(0);
            }
            if(strcmp(input, "exit\n") == 0)
                exit(0);
        }
        input[strlen(input)-1] = '\0';
        token = strtok(input, " ");
        args[0] = token;
        i = 1;
        while(token != NULL) {
            token = strtok(NULL, " ");
            args[i] = token;
            i++;
        }
        pid = fork();
        if(pid < 0) {
            fprintf(stderr, "Fork failed");
            exit(-1);
        }
        else if(pid == 0) {
            execvp(input, args);
            exit(0);
        }
        else {
            wait(0);
        }
    }
    return 0;
}
```

```

u17041@solgae:~/prog$ ./a.out
mysh > exit
u17041@solgae:~/prog$ ls
1-1.c 1-2-1.c 1-2-2.c 2.c 3.c a.out
u17041@solgae:~/prog$ ./a.out
mysh > whoami
u17041
mysh > id
uid=1189(u17041) gid=1001(u_student) groups=1001(u_student)
mysh > date
2021. 04. 16. (금) 22:55:39 KST
mysh > finger u17041
Login: u17041 Name: ? -^YM-^M? -^D갓M-^Z?
Directory: /home/u17041 Shell: /bin/bash
On since Fri Apr 16 21:35 (KST) on pts/9 from 110.10.226.34
No mail.
No Plan.
mysh > ls -s
total 40
 4 1-1.c  4 1-2-1.c  4 1-2-2.c  4 2.c  4 3.c 20 a.out
mysh > ls -l
total 40
-rw-r--r-- 1 u17041 u_student 683 4월 16 22:05 1-1.c
-rw-r--r-- 1 u17041 u_student 894 4월 16 22:22 1-2-1.c
-rw-r--r-- 1 u17041 u_student 492 4월 16 22:29 1-2-2.c
-rw-r--r-- 1 u17041 u_student 763 4월 16 22:53 2.c
-rw-r--r-- 1 u17041 u_student 703 4월 16 22:35 3.c
-rwxr-xr-x 1 u17041 u_student 17168 4월 16 22:54 a.out
mysh > exit
u17041@solgae:~/prog$ █

```

exit와 ctrl+D명령어에 잘 종료된다.

3.

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <string.h>

#define BUFFER_SIZE 25
#define READ_END 0
#define WRITE_END 1

int main(void)
{
    char write_msg[BUFFER_SIZE] = "Greetings";
    char read_msg[BUFFER_SIZE];
    pid_t pid;
    int fd[2];

    if (pipe(fd) == -1) {
        fprintf(stderr, "Pipe failed");
        return 1;
    }

    pid = fork();
    if (pid < 0) {
        fprintf(stderr, "Fork failed");
        return 1;
    }
    if (pid > 0) {
        close(fd[READ_END]);
        write(fd[WRITE_END], write_msg, strlen(write_msg)+1);
        close(fd[WRITE_END]);
    }
    else {
        close(fd[WRITE_END]);
        read(fd[READ_END], read_msg, BUFFER_SIZE);
        printf("child read %s\n", read_msg);
        close(fd[READ_END]);
    }

    return 0;
}
```

"3.c" [새로운] 41L, 703C 저장 했습니다 41,0-1 모두

```
u17041@solgae:~/prog$ ./a.out
child read Greetings
u17041@solgae:~/prog$ ls
1-1.c 1-2-1.c 1-2-2.c 3.c a.out
u17041@solgae:~/prog$
```

파이프는 부모프로세스에서 자식 프로세스로 Greetings 메시지를 보내는 역할을 한다.