

KU_IPC 보고서



과목	임베디드 소프트웨어1
학과	컴퓨터 공학과
학번	201311299
이름	이원오

<Submission>

- Sys V Message Queue 시스템 구현 (커널 모듈을 통해서 메시지 큐 구현.)
- 기본적인 함수들, msgget, msgclose, msgsnd, msgrcv를 구현.
- 커널 영역 내에서 메시지 큐를 생성하고 활용.
- 메시지 큐는 유저로부터 key값과 msgflg의 값에 따라서 그에 맞게 생성 (key는 해당 메시지 큐를 인식하는 고유값.)
- spin_lock을 통해서 프로세스간의 메시지 큐 사용에 발생할 수 있는 동기화 문제 해결.

<Basic Design>

1. ku_ipc_lib.c

```
#include <stdio.h>
#include <sys/fcntl.h>
#include <sys/ioctl.h>
#include <stdlib.h>
#include <string.h>

#include "ku_ipc.h"

#define DEV_NAME "ku_ipc_dev"

#define IPC_START_NUM 0x80
#define IPC_NUM1 IPC_START_NUM+1
#define IPC_NUM2 IPC_START_NUM+2
#define IPC_NUM3 IPC_START_NUM+3
#define IPC_NUM4 IPC_START_NUM+4
#define IPC_NUM5 IPC_START_NUM+5
#define IPC_NUM6 IPC_START_NUM+6
#define IPC_NUM7 IPC_START_NUM+7
#define IPC_NUM8 IPC_START_NUM+8
#define IPC_NUM9 IPC_START_NUM+9
#define SIMPLE_IPC_NUM 'z'

#define KU_IPC_SEND _IOWR(SIMPLE_IPC_NUM, IPC_NUM1, unsigned long *)
#define KU_IPC_RECEIVE _IOWR(SIMPLE_IPC_NUM, IPC_NUM2, unsigned long *)
#define KU_IPC_MSGGET _IOWR(SIMPLE_IPC_NUM, IPC_NUM3, unsigned long *)
#define KU_IPC_MSGCLOSE _IOWR(SIMPLE_IPC_NUM, IPC_NUM4, unsigned long *)
#define KU_IPC_EXIST_KEY _IOWR(SIMPLE_IPC_NUM, IPC_NUM5, unsigned long *)
#define KU_IPC_CREATE_QUEUE _IOWR(SIMPLE_IPC_NUM, IPC_NUM6, unsigned long *)
#define KU_IPC_FULL_QUEUE _IOWR(SIMPLE_IPC_NUM, IPC_NUM7, unsigned long *)
#define KU_IPC_NO_QUEUE _IOWR(SIMPLE_IPC_NUM, IPC_NUM8, unsigned long *)
#define KU_IPC_EMPTY_QUEUE _IOWR(SIMPLE_IPC_NUM, IPC_NUM9, unsigned long *)

struct msgbuf{
    long type;
    char text[100];
};

struct ipcbuf{
    int msqid;
    void *msgp;
    int msgsz;
    long msgtyp;
    int msgflg;
};
```

- 필요한 헤더들을 추가
- 전처리를 통해서 ioctl에 대한 command 설정 (해당 기능에 따라서 9개 설정)
- 메시지 큐에 적재할 메시지 버퍼 구조체 선언 및 ipc함수에 대한 인자값을 담은 구조체 선언.

```
int ku_msgget(int key, int msgflg);
int ku_msgclose(int msqid);
int ku_msgsnd(int msqid, void *msgp, int msgsz, int msgflg);
int ku_msgrcv(int msqid, void *msgp, int msgsz, long msgtyp, int msgflg);
```

- 라이브러리에 포함된 사용자가 실질적으로 호출해서 쓸 수 있는 함수들.

2. 전처리 (ku_ipc.c)

```
spinlock_t lock;

struct msgbuf{
    long type;
    char text[100];
};

struct ipcbuf{
    int msqid;
    void *msgp;
    int msgsz;
    long msgtyp;
    int msgflg;
};

struct q{
    struct list_head list;
    struct msgbuf kern_buf;
};

struct queues{
    struct list_head list;
    struct q kern_q;
    int key;
};

struct queues kern_queue;
```

- ipcbuf : 메시지를 받을 때 그리고 보낼 때 필요한 변수들을 포함한 구조체.
- queues : 전체적인 메시지 큐들을 관리하는 Kernel Linked List 구조체.
- q : 메시지 버퍼를 멤버로 갖는 Kernel Linked List 구조체.
- msgbuf : 사용자가 실질적으로 사용하는 메시지의 형식 구조체.
- spinlock_t lock : 동기화 문제를 해결할 spinlock 형식의 전역변수.

<Description for important functions>

● ku_ipc_lib.c

1. int ku_msgget(int key, int msgflg)

int ku_msgget (int key, int msgflg)	Functionality	해당 key값을 갖는 메시지 큐 생성
	Parameters	식별자인 key, flag값인 msgflg (IPC_CREAT일 경우는 식별자를 반환하고 IPC_EXCL일 경우 동일한 key를 갖는 메시지 큐가 존재할 경우 -1리턴)
	Return Value	성공했을 시 해당 메시지 큐의 msqid값. 실패했을 시 -1 리턴

2. int ku_msgclose(int msqid)

int ku_msgclose (int msqid)	Functionality	해당 msqid값을 갖는 메시지 큐를 연결 해제
	Parameters	msqid (메시지 큐를 식별하는 식별자)
	Return Value	성공시 0, 실패시 -1 리턴

3. int ku_msgsnd(int msqid, void* msgp, int msgsz, int msgflg)

int ku_msgsnd (int msqid, void *msgp, int msgsz, int msgflg)	Functionality	msqid에 해당하는 큐에 데이터 전송
	Parameters	msqid : 메시지 큐 id, msgp : 메시지 큐로 보낼 데이터가 있는 버퍼의 시작 주소, msgsz : 메시지큐로 보낼 데이터의 크기, msgflg : 메시지의 flag(IPC_NOWAIT의 경우 해당 메시지 큐의 공간이 없을 때는 메시지 전송하지 않고 즉시 리턴하고 0일 경우는 메시지큐의 공간이 생길 때까지 대기)
	Return Value	성공 시 0, 실패 시 -1

4. int ku_msgrcv(int msqid, void *msgp, int msgsz, long msgtyp, int msgflg)

int ku_msgrcv (int msqid, void *msgp, int msgsz, long msgtyp, int msgflg)	Functionality	msqid에 해당하는 큐로부터 데이터 수신
	Parameters	msqid : 메시지 큐 ID, msgp : 데이터를 수신할 버퍼의 시작주소, msgsz : 큐로부터 받을 데이터의 크기, msgtyp : 메시지의 형식, msgflg : 메시지 flag(IPC_NOWAIT(메시지 큐에 메시지 없을 경우)이면 즉시 리턴, 아닐 경우 메시지에 메시지가 있을 때까지 대기, MSG_NOERROR(큐에 있는 자료가 msgsz보다 큰 경우)이면 읽을 수 있는 만큼만 수신, 아닌 경우는 자료가 있어도 -1 리턴)
	Return Value	성공 시 수신된 데이터의 길이 실패 시 -1 리턴

● ku_ipc.c

1. int is_key(int *key)

int is_key (int *key)	Functionality	key값을 갖는 큐가 리스트에 존재하는 지 확인하는 함수
	Parameters	찾고 싶은 key 값의 포인터
	Return Value	해당 키를 갖는 큐가 존재할 경우 해당 큐의 key값을, 존재하지 않는다면 0리턴

2. int make_queue(int *key)

int make_queue (int *key)	Functionality	key값을 갖는 큐를 전체 리스트에 추가해주는 함수.
	Parameters	추가하고 싶은 큐의 key값
	Return Value	해당 key의 값 리턴

3. int remove_queue(int *msqid)

int remove_queue (int *msqid)	Functionality	msqid값을 갖는 해당 큐를 리스트에서 제거
	Parameters	제거하고 싶은 큐의 msqid값.
	Return Value	삭제 성공 시 0 리턴 해당 msqid값을 갖는 큐가 존재하지 않을 시 -1 리턴

4. struct q* get_queue(int key)

struct q* get_queue (int key)	Functionality	해당 key를 갖는 큐를 반환
	Parameters	찾고 싶은 큐의 key값
	Return Value	성공 시 해당 key를 갖는 구조체 q를 반환 실패 시 0을 리턴

5. int is_full(int *msqid)

int is_full (int *msqid)	Functionality	해당 msqid를 갖는 큐가 꽉 찼는지 확인해주는 함수
	Parameters	찾고 싶은 큐의 msqid의 포인터
	Return Value	해당 msqid값을 갖는 메시지 큐의 size가 KUIPC_MAXVOL(한계치)사이즈보다 크거나 큐의 개수가 KUIPC_MAXMSG보다 큰 경우 1, 아니면 0 리턴

6. int is_empty(int *msqid)

int is_empty (int *msqid)	Functionality	해당 msqid를 갖는 큐가 비어있는지 확인해주는 함수
	Parameters	찾고 싶은 큐의 msqid의 포인터
	Return Value	해당 msqid를 갖는 큐의 메시지 개수가 0이라면 1 리턴, 아니라면 0 리턴

7. int get_queue_size(int msqid)

int get_queue_size (int msqid)	Functionality	해당 msqid의 큐의 데이터의 총 크기를 구하는 함수
	Parameters	찾고 싶은 큐의 msqid값
	Return Value	인자 msqid에 해당하는 큐의 데이터 사이즈(byte단위), 없으면 0 리턴

8. int get_num(int msqid)

int get_num (int msqid)	Functionality	해당 msqid를 갖는 큐의 메시지 개수를 확인하는 함수
	Parameters	찾고 싶은 큐의 msqid값
	Return Value	msqid에 해당하는 큐의 메시지 개수를 리턴, 없을 경우 0리턴

9. int no_queue(int *msqid)

int no_queue (int *msqid)	Functionality	해당 msqid를 갖는 큐의 존재를 확인하는 함수
	Parameters	찾고 싶은 큐의 msqid의 포인터
	Return Value	큐가 존재하지 않을 시 1 리턴, 존재하는 경우 0 리턴

10. static int read_ipc(struct ipcbuf *ipc)

static int read_ipc (struct ipcbuf *ipc)	Functionality	ku_msgrcv의 중심함수로서, 사용자가 입력한 인자값(ipcbuf 구조체)의 조건에 맞는 큐에서 데이터를 커널영역에서 user영역으로 복사(copy_to_user함수 사용)하고 그 복사된 데이터만큼의 크기를 구하는 함수
	Parameters	사용자가 입력한 인자값(ipcbuf 구조체)
	Return Value	해당 큐에 큐가 없을 경우 0 리턴, msqid에 해당하는 큐가 없을 경우는 -1 리턴, 성공 시에는 읽은 데이터 크기 리턴

11. static int write_ipc(struct ipcbuf *ipc)

static int write_ipc (struct ipcbuf *ipc)	Functionality	ku_msgsnd의 중심함수로서 사용자가 입력한 인자값(ipcbuf 구조체)의 조건(msqid값에 맞는 큐)에 맞게 데이터를 유저영역에서 커널 영역으로 복사(copy_from_user함수 사용)하는 함수
	Parameters	사용자가 입력한 인자값(ipcbuf 구조체)
	Return Value	msqid에 해당하는 큐가 없을 경우는 -1 리턴, 성공 시에는 복사 실패한 데이터 크기 리턴

12. static long ku_ipc_ioctl

(struct file *file, unsigned int cmd, unsigned long arg)

static long ku_ipc_ioctl (struct file *file, unsigned int cmd, unsigned long arg)	Functionality	캐릭터 디바이스에 등록할 ioctl 함수 cmd(커맨드)에 들어갈 값에 따라서 각기 다른 기능 실행
	Parameters	file descriptor (open한 디바이스의 정보), ioctl에 쓰일 명령어, ioctl함수에 필요한 인자값
	Return Value	cmd(명령)에 따라서 다양한 리턴 값

13. static int __init ku_ipc_init(void)

static int __init ku_ipc_init (void)	Functionality	커널 모듈을 시작하는 함수 캐릭터 디바이스 등록 및 커널영역에 메시지 큐 리스트 초기화
	Parameters	-
	Return Value	성공 시 0 반환

14. static void __exit ku_ipc_exit(void)

static void __exit ku_ipc_exit (void)	Functionality	커널 모듈을 종료하는 함수 캐릭터 디바이스 해제 및 커널 영역에 존재하는 메시지 큐 리스트를 삭제
	Parameters	-
	Return Value	-