

## ku\_ipc.c 구조체

0	Structure	Description	
		Members	

1	Structure	Description	sysv ipc 메시지 버퍼
	<pre>struct ku_msgbuf {     long type;     char text[MSG_LEN]; };</pre>	Members	<p>type : sysv type.</p> <p>text : sysv mtext</p>

2	Structure	Description	msgget ,msgclose, msgrcv, msgsnd 에서 ioctl로 보낼 데이터를 구조체로 묶음
	<pre>struct msgq_metadata {     int msqid;     void *msgp;     int msgsz;     long msgtyp;     int msgflg; };</pre>	Members	Desc에 설명되어있는 대로

3	Structure	Description	커널의 연결리스트에서 메시지 버퍼끼리의 link를 위해 사용하기 위한 listnode.
	<pre>struct ku_listnode {     struct ku_msgbuf *msg;     struct list_headlist; };</pre>	Members	

4	Structure	Description	커널에서 process의 reference counter를 추적하기 위한 pid listnode.
	<pre>struct ku_pid_listnode { int pid; struct list_head list; };</pre>	Members	pid : 현재 사용중인 pid

5	Structure	Description	각 메시지 큐의 entry와 그 entry에 관한 정보를 관리하기 위한 structure
	<pre>struct msgq_wrapper { int msgq_ref_count[MAX_ENTRY]; int msgq_bytes[MAX_ENTRY]; int msgq_num[MAX_ENTRY]; struct ku_listnode msgq_entry[MAX_ENTRY]; struct ku_pid_listnode msgq_entry_pid[MAX_ENTRY]; };</pre>	Members	<p>msgq_ref_count : 각 entry별 reference counter</p> <p>msgq_bytes : 각 entry별 저장된 메시지의 bytes</p> <p>msgq_num : 각 entry별 저장된 메시지의 개수</p> <p>msgq_entry : 각 entry별 entry listhead의 시작주소</p> <p>msgq_entry_pid : 각 entry별 entry를 사용하고 있는 pid의 정보를 담고있는 listhead의 시작주소</p>

# ku\_ipc\_lib.c 함수

0	FUNCTION	Description	
		Parameter	
		Return Value	
1	FUNCTION	Description	"라이브러리로 msgid를 전달한 process"가 같은 msgid를 사용하고 있었는지 확인
	static int is_using_msgq(int msgid, int pid)	Parameter	msgid : 전달받은 msgid pid : 라이브러리로 msgid를 전달한 프로세스의 pid
		Return Value	사용하고 있으면 1 사용하고 있지 않았으면 0
2	FUNCTION	Description	msgid의 메시지큐를 사용하고 있는 pid를 reference counter를 관리하는 list에 추가
	static int add_pid_to_list(int msgid, int pid)	Parameter	msgid : 전달받은 msgid pid : 라이브러리로 msgid를 전달한 프로세스의 pid
		Return Value	1
3	FUNCTION	Description	커널 내부에서 msgget 로직을 수행하는 함수. Msgid 범위 확인, 전달받은 msgq를 이미 사용하고 있었는지 확인. KU_IPC_CREAT 와 KU_IPC_EXCL에 따라 어떤 값을 리턴할지 판단.
	static int ku_ipc_msgget_ioctl(unsigned long arg)	Parameter	arg : unlocked_ioctl으로 받은 인자를 넘겨받는다. 주로 msgq의 metadata를 받는다.

		Return Value	<p>msgid 범위를 벗어남, msgq를 이미 사용하고 있었을 때 -1 리턴</p> <p>1. 플래그가 없는 경우 (msgflg = 0) key값에 해당하는 message queue가 존재하는 경우 Queue ID 반환 key값에 해당하는 message queue가 존재하지 않는 경우 -1 반환</p> <p>2. IPC_CREAT 만 사용하는 경우 (msgflg = IPC_CREAT) key 값에 해당하는 message queue가 있을 경우 해당 queue에 대한 Queue ID 반환 key 값에 해당하는 message queue가 없을 경우 message queue 생성 후 Queue ID 반환</p> <p>3. IPC_CREAT   IPC_EXCL 인 경우 (msgflg = IPC_CREAT   IPC_EXCL) key 값에 해당하는 message queue가 존재하는 경우 -1 반환 key 값에 해당하는 message queue가 없는 경우 message queue 생성 후 Queue ID 반환</p>
4 FUNCTION		Description	msgclose에서 성공적으로 msgq가 닫혔다면, reference count를 줄이고, reference를 사용하는 pid를 list에서 제거한다.
	static int remove_pid_from_list(int msgid, int pid)	Parameter	<p>msgid : 전달받은 msgid</p> <p>pid : 라이브러리로 msgid를 전달한 프로세스의 pid</p>
		Return Value	1

5	FUNCTION	Description	<p>커널 내부에서 msgclose 로직을 수행하는 함수.</p> <p>Msgid 범위 확인, 전달받은 msgq를 이미 사용하고 있었는지 확인.</p> <p>성공적인 함수 실행 시 0 리턴.</p> <p>사용중이 아니었던 msgq라면 -1 리턴.</p>
	static int ku_ipc_msgclose_ioctl(unsigned long arg)	Parameter	<p>arg : unlocked_ioctl으로 받은 인자를 넘겨받는다.</p> <p>주로 msgq의 metadata를 받는다.</p>
		Return Value	<p>성공적인 함수 실행 시 0 리턴.</p> <p>사용중이 아니었던 msgq라면 -1 리턴.</p>
6	FUNCTION	Description	<p>메시지 큐가 꽉 차있거나, 더 이상 byte를 추가로 받아 list에 추가 할 수 없을 때</p> <p>1을 리턴.</p> <p>메시지 큐를 받을 수 있는 상황(위 조건의 부정)이라면 0을 리턴.</p>
	static int is_blocked_condition(int msgid, int msgsz)	Parameter	<p>msize : 메시지 사이즈</p> <p>msgid : 메시지 entry id</p>
		Return Value	<p>메시지 큐가 꽉 차있거나, 더 이상 byte를 추가로 받아 list에 추가 할 수 없을 때</p> <p>1을 리턴.</p> <p>메시지 큐를 받을 수 있는 상황(위 조건의 부정)이라면 0을 리턴.</p>
7	FUNCTION	Description	<p>명시한 msgq의 entry에 메시지를 넣는 함수.</p> <p>메시지 큐가 꽉 차있거나, 더 이상 byte를 추가로 받아 list에 추가 할 수 없을 때 wait_event_interruptible 실행.</p> <p>메시지를 넣은 후 rcv에 의해 blocking되어있던 프로세스를 깨운다.</p>

	static int ku_ipc_msgsnd_ioctl(unsigned long arg)	Parameter	msgq의 metadata(의 주소)를 받는다.
		Return Value	Length 를 넘어서 큐에 메시지를 넣을 수 없는 경우 flag 값이 KU_IPC_NOWAIT이라면 안기다리고, 안넣고 ,return -1 KU_IPC_NOWAIT 없이 그냥 msg==0이라면, 해당 메시지 큐가 꽉 찼다면 BLOCKING 으로 재워서 available 할 때 blocking 되었던 프로세스를 깨워서 0을 리턴.
8	FUNCTION	Description	
	static int ku_ipc_msgrcv_ioctl(unsigned long arg)	Parameter	
		Return Value	메시지큐에 수신한 메시지가 있을 때까지 대기하고 수신할 메시지가 있다면 그때 BLOCKING 에서 깨우고 리턴 msgsz  KU_IPC_NOWAIT : 만약 메시지 큐에서 가져올 메시지가 없다면 즉시 -1 리턴  KU_MSG_NOERROR : caller function이 명시한 msgsz보다 메시지 큐의 msg가 크다면 truncate 하고 msgsz를 리턴
9	FUNCTION	Description	cdev의 file_operations에 전달해줄 unblocked_ioctl함수
	static long ku_ipc_ioctl(struct file *file, unsigned int cmd, unsigned long arg)	Parameter	-
		Return Value	각 subfunction에서 반환한 값
11	FUNCTION	Description	사용할 datastructure를 초기화시킨다. cdev, spinlock, waitqueue, struct wrapper를 초기화
	static int init_datastructure(void) static void init_msgq_wrapper(void) static void init_waitqueue(void) static void init_spinlock(void) static int init_cdev(void)	Parameter	-
		Return Value	1

10	FUNCTION	Description	지금까지 사용한 동적할당된 datastructure의 메모리를 반환한다. 주로, list관련 메모리
	static void delete_datastructure(void)	Parameter	-
		Return Value	-

## ku\_ipc\_lib.c 구조체

0	Structure	Description	
		Members	

1	Structure	Description	sysv ipc 메시지 버퍼
	<pre>struct ku_msgbuf {     long type;     char text[MSG_LEN]; };</pre>	Members	<p>type : sysv type.</p> <p>text : sysv mtext</p>

2	Structure	Description	msgget ,msgclose, msgrcv, msgsnd 에서 ioctl로 보낼 데이터를 구조체로 묶음
	<pre>struct msgq_metadata {     int msqid;     void *msgp;     int msgsz;     long msgtyp;     int msgflg; };</pre>	Members	Desc에 설명되어있는 대로



--	--	--	--



--	--	--	--



# ku\_ipc\_lib.c 함수

0	FUNCTION	Description	
		Parameter	
		Return Value	
1	FUNCTION	Description	ioctl을 call 할 라이브러리 함수
	int ku_msgget(int key, int msgflg)	Parameter	
		Return Value	ioctl이 리턴하는 값
2	FUNCTION	Description	ioctl을 call 할 라이브러리 함수
	int ku_msgclose(int msqid)	Parameter	msgid : 메시지큐의 id
		Return Value	ioctl이 리턴하는 값
3	FUNCTION	Description	ioctl을 call 할 라이브러리 함수
	int ku_msgsnd(int msqid, void *msgp, int msgsz, int msgflg)	Parameter	msgid : 메시지큐의 id msgp : 메시지버퍼의 시작주소 msgsz : text(mtext)의 길이/크기 msgflg : 메시지 플래그
		Return Value	ioctl이 리턴하는 값
4	FUNCTION	Description	ioctl을 call 할 라이브러리 함수
	int ku_msgrcv(int msqid, void *msgp, int msgsz, long msgtyp, int msgflg)	Parameter	msgid : 메시지큐의 id msgp : 메시지버퍼의 시작주소 msgsz : text(mtext)의 길이/크기 msgtyp : sysv ipc의 msgtype msgflg : 메시지 플래그
		Return Value	ioctl이 리턴하는 값

