



임베디드 소프트웨어 과제 #1

KU_IPC

201511291 컴퓨터공학과 장유준



1. struct

Struct params	members	
		Int msqid, msgsz, msgflg Void *msgp Long msgtyp

- 유저가 ioctl 함수로 보낼 인자를 담을 구조체

1. struct

Struct msg_queue	members	<pre>struct list_head list int ref_counter int msg_vol int msg_counter struct msgbuf msg</pre>
-------------------------	---------	------------------------------------------------------------------------------------------------

- 커널 내부의 msg queue 정의
- 메시지 큐를 사용하는 프로세스의 개수를 ref_counter로 정의
- 메시지의 개수를 msg_counter, 메시지가 차지하는 메모리 양을 msg_vol 으로 정의

1. struct

Struct file_operations ku_ipc_fops	members	.open = ku_ipc_open, .release = ku_ipc_release, .unlocked_ioctl = ku_ioctl,
---------------------------------------	---------	-----------------------------------------------------------------------------------

- Fops 구조체 내부의 함수 pointer 연결
- 유저가 ioctl() 함수 호출 시 커널 내부의 ku_ioctl 함수 호출

2. functions

delay()	Functionality	Delay time
	Parameters	Int sec
	Return Value	Void

- Sec 초 만큼 waiting 하는 기능
- Ctrl + C / interrupt 로 함수 중단 불가

2. functions

kernel_get()	Functionality	Get queue
	Parameters	Int key, int msgflg
	Return Value	int

- IOCTL 의 KU_MSGGET 으로 호출
- 해당 key 값의 queue호출 , 사용중인지 판별 후 해당 queue의 msqid값 반환
- Msgflg 인자값이 KU_IPC_CREAT인 경우 해당 queue의 msqid값 반환 후 종료
- KU_IPC_EXCL 인 경우 사용 중일 경우 -1반환, 사용 중이 아닌 경우 msqid값 반환 후 종료
- Reference counter 증가, 감소를 통해 queue에 연결된 프로세스 개수 조정

2. functions

kernel_close()

Functionality

Close queue

Parameters

Int msqid

Return Value

int

- IOCTL 의 KU_MSGCLOSE 으로 호출
- Msqid 값을 통해 해당 queue의 사용 여부 판단 후 종료 여부 결정
- 사용 중이 아닌 queue의 msqid를 받았을 때 -1 반환
- 정상 종료 된 경우 0 반환

2. functions

kernel_snd()	Functionality	receive a message to kernel
	Parameters	int msqid, void *msgp, int msgsz, int msgflg
	Return Value	int

- IOCTL 의 KU_MSGSND로 호출
- Msqid로 얻어진 kernel 내부의 queue에 메시지를 inqueue 하는 기능
- 파라미터로 들어온 user_buffer에 저장된 메시지를 copy_from_user 함수를 통해 kernel내부의 msg queue에 inqueue
- Message counter와 message volume을 MAX 값과 비교해 inqueue 가능 여부 판단
- KU_IPC_NOWAIT flag를 통해 queue의 사이즈보다 크게 입력 하는 경우 , 오류 또는 대기 설정 가능 , 대기 할 시 함수 polling을 통해 대기
- 성공적인 message send 인 경우 0반환, 실패 시 -1 반환

2. functions

kernel_rcv()	Functionality	Copy a message to user
	Parameters	int msqid, void *msgp, int msgsz, long msgtyp, int msgflg
	Return Value	int

- IOCTL 의 KU_MSGRCV로 호출
- 유저가 요청한 msqid를 이용해 kernel 내부의 msgqueue에 있는 메시지를 copy_to_user 함수를 이용해 user_buffer로 전달
- Msg queue 안에 msg가 없는 경우 KU_IPC_NOWAIT flag 를 통해 대기 또는 즉시 실패 반환 설정 가능
- 함수의 polling을 통한 receive 대기
- User가 receive를 원하는 msg의 size가 queue에 있는 msg보다 큰 경우 KU_MSG_NOERROR flag를 통해 메시지 컷팅 또는 실패 반환 설정 가능
- Msgtyp 변수를 통해 원하는 type의 메시지만 수신 가능

2. functions

ku_ioctl()	Functionality	ioctl 함수 기능 정의
	Parameters	struct file *file, unsigned int cmd, unsigned long arg
	Return Value	long

- 유저가 ku_msgget() 함수 호출 시 kernel_get() 함수 호출
- ku_msgclose() 함수 호출 시 kernel_close() 함수 호출
- ku_msgsnd() 함수 호출 시 kernel_snd() 함수 호출
- ku_msgrcv() 함수 호출 시 kernel_rcv() 함수 호출
- User가 arg 인자로 넘겨준 값을 통해 kernel 내부의 함수의 파라미터로 전달

3. Flow (write)

User (writer)

(1) `ku_msgget()`를 통해 msgqueue id 획득

(3) `ku_msgsnd()`를 사용해 획득한 id의 queue에 원하는 msg 전달 요청

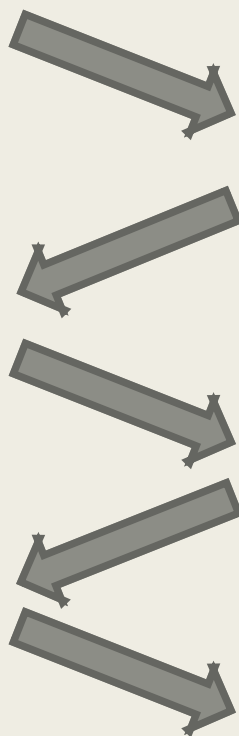
(5) `ku_close()`를 통해 msgqueue 연결 해제 요청

Kernel

(2) `kernel_get()` 함수 호출, reference counter 증가, 유저에게 msgqueue id 반환

(4) `kernel_snd()` 함수 호출, 커널 내부의 msg queue에 msg inqueue

(6) `kernel_close()` 호출, 해당 queue의 reference counter 감소, 연결 해제



3. Flow (receive)

User (receiver)

(1) ku_msgget()를 통해 msgqueue id 획득

(3) ku_msgrcv()를 사용해 획득한 id의 queue의 msg receive 요청

(5) ku_close()를 통해 msgqueue 연결 해제 요청

Kernel

(2) kernel_get() 함수 호출, reference counter 증가, 유저에게 msgqueue id 반환

(4) kernel_rcv() 함수 호출, 커널 내부의 msgqueue에서 유저에게 msg 전달

(6) kernel_close() 호출, 해당 queue의 reference counter 감소, 연결 해제

