

# Message queue

2021.05.20

Sang-woo Lee

glutton.leesw@gmail.com



# Contents

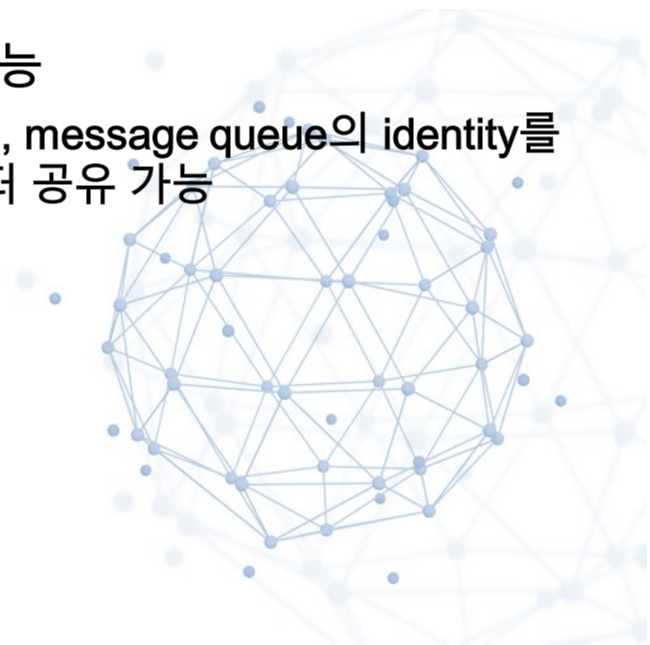
---

- Message queue



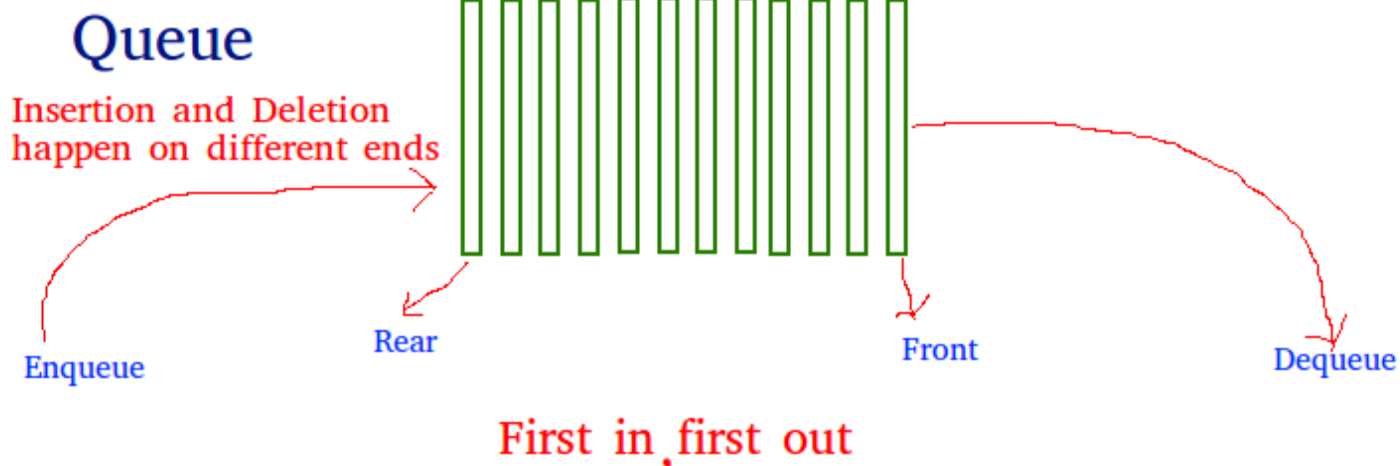
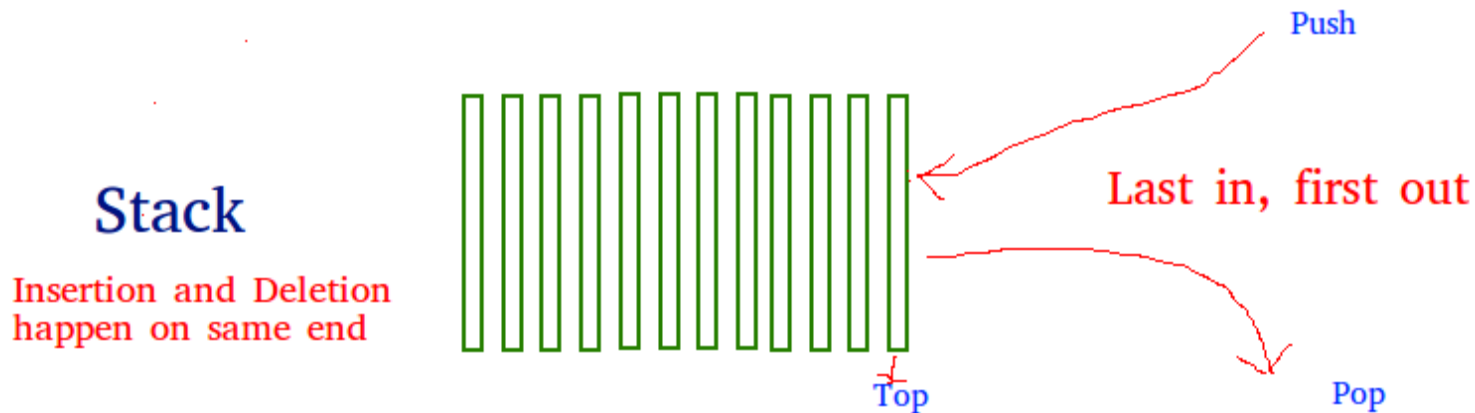
# Message queue

- Queue
  - 컴퓨터의 기본적인 자료 구조의 한 가지
  - 먼저 들어간 데이터가 먼저 나오는 FIFO (First In First Out) 구조로 저장하는 형식
  - Stack과 반대되는 개념
- Message queue
  - Message를 자료 구조의 한 가지인 queue로 관리
  - 커널에서 전역적으로 관리되며, 모든 프로세스에서 접근 가능
    - ✓ 하나의 message queue 서버가 커널에 요청해 메시지 큐 작성 시, message queue의 identity를 아는 모든 프로세서가 동일한 message queue에 접근하여 데이터 공유 가능
  - 사용 방법이 매우 직관적이고 간단함



# Message queue

- Queue & stack 자료 구조



# Message queue

- Message queue create
  - 임의의 프로세스가 message queue를 생성

유형	내용
헤더	<pre>#include &lt;sys/types.h&gt; #include &lt;sys/ipc.h&gt; #include &lt;sys/msg.h&gt;</pre>
함수	<pre>int msgget (key_t key, int msgflg);</pre>
설명	Message queue 생성
매개변수	key : 시스템에서 다른 큐와 구별되는 번호 msgflg : 옵션
반환 값	성공: message queue 식별자 실패: -1

- Message queue create option

msgflg	내용
IPC_CREAT	key에 해당하는 큐가 있다면 큐의 식별자를 반환하고, 없다면 큐를 생성
IPC_EXCL	key에 해당하는 큐가 없다면 큐를 생성하고, 큐가 있다면 -1을 반환

# Message queue

- Message send

- 프로세스 A가 message queue에 자신이 생성한 메시지 전송
- Message queue는 전송 받은 메시지를 자신의 저장 공간에 저장
- 프로세스로부터 연속적으로 수신되는 메시지는 연결 리스트로 계속해서 저장됨

유형	내용
헤더	<code>#include &lt;sys/types.h&gt;</code> <code>#include &lt;sys/ipc.h&gt;</code> <code>#include &lt;sys/msg.h&gt;</code>
함수	<code>int msgsnd (int msqid, const void * msgp, size_t msgsz, int msgflg);</code>
설명	Message queue에 message 전송
매개변수	msqid : message queue 식별자 msgp : 전송할 자료 msgsz : 전송할 자료의 크기 msgflg : 동작 옵션
반환 값	성공: 0 실패: -1



# Message queue

- Message send option

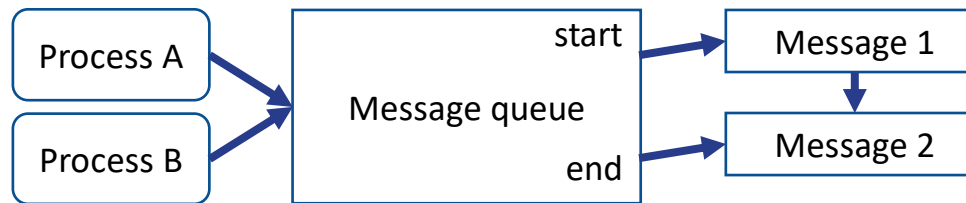
msgflg	내용
0	큐에 공간이 생길 때까지 대기
IPC_NOWAIT	큐에 여유 공간이 없다면 바로 -1로 복귀



# Message queue

- Message queue access

- 다른 프로세스가 message queue에 있는 데이터를 가지고 오기 위해서는 해당 message queue에 접근해야 함



- Message receive

유형	내용
헤더	<code>#include &lt;sys/types.h&gt;</code> <code>#include &lt;sys/ipc.h&gt;</code> <code>#include &lt;sys/msg.h&gt;</code>
함수	<code>int msgrcv (int msqid, const void * msgp, size_t msgsz, long msgtyp, int msgflg);</code>
설명	Message 수신
매개변수	msqid : message queue 식별자 msgp : 전송받을 자료 msgsz : 전송받을 자료의 크기 msgflg : 동작 옵션
반환 값	성공: 0 실패: -1



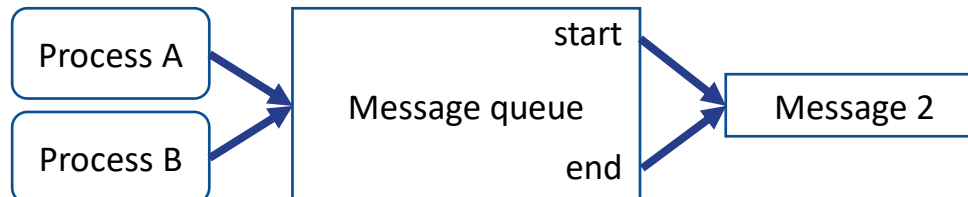
# Message queue

- Message receive option

- 프로세스 B가 메시지 수신 시, 기본적으로 첫 번째 메시지를 수신해야 함
- 수신된 메시지는 큐에서 삭제됨

msgtyp	내용
0	큐에 자료가 있다면 첫 번째 자료를 읽어 들임
양수	양수로 지정한 값과 같은 data_type의 자료 중 첫 번째 자료를 읽어 들임
음수	음수 값을 절대값으로 변경하고, 이 절대값과 같거나 보다 제일 작은 data_type의 자료를 읽어 들임 message queue에 data_type이 1, 5, 15이고, -10을 지정했다면 1의 데이터를 읽어 들임

msgflg	내용
IPC_NOWAIT	큐에 자료가 없다면 기다리지 않고 -1로 복귀
MSG_NOERROR	큐에 있는 자료가 준비된 크기보다 크다면, 초과되는 부분을 잘라내고 읽어 들일 수 있는 부분만 가져옴 이 옵션이 없을 경우, 큐에 자료가 있더라도 -1로 실패함



# Message queue

- Message queue control

유형	내용
헤더	<code>#include &lt;sys/types.h&gt;</code> <code>#include &lt;sys/ipc.h&gt;</code> <code>#include &lt;sys/msg.h&gt;</code>
함수	<code>int msgctl (int msqid, int cmd, struct msqid_ds * buf);</code>
설명	Message queue state control
매개변수	msqid : message queue 식별자 cmd : 제어 명령 buf : message queue 자료를 받을 버퍼
반환 값	성공: 0 실패: -1

- Message queue control

cmd	내용
IPC_STAT	큐의 현재 상태를 buf에 저장
IPC_SET	큐의 상태를 buf 값으로 변경 모든 정보는 저장할 수 없으며, msg_perm과 msg_qbytes 내용만 변경 가능
IPC_RMID	큐를 삭제함 큐 삭제 시, 버퍼가 필요 없으므로 buf를 0으로 지정

# Message queue

- msq\_sender1.c

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

struct real_data{
    short age;
    char name[16];
};

struct message{
    long msg_type;
    struct real_data data;
};
```

```
void printMsgInfo(int msqid){
    struct msqid_ds m_stat;
    printf("===== messege queue info =====\n");
    if(msgctl(msqid,IPC_STAT,&m_stat)==-1){
        printf("msgctl failed");
        exit(0);
    }
    printf(" message queue info \n");
    printf(" msg_lspid : %d\n",m_stat.msg_lspid);
    printf(" msg_qnum : %d\n",m_stat.msg_qnum);
    printf(" msg_stime : %d\n",m_stat.msg_stime);
    printf("===== messege queue info end =====\n");
}
```

# Message queue

- msq\_sender1.c

```
int main(){
    key_t key=12345;
    int msqid;
    struct message msg;
    msg.msg_type=1;
    msg.data.age=20;

    strcpy(msg.data.name,"UNKNOWN");
    //msqid를 얻어옴.
    if((msqid=msgget(key,IPC_CREAT|0666))== -1){
        printf("msgget failed\n");
        exit(0);
    }
    //메시지 보내기 전 msqid_ds를 한번 보자.
    printMsgInfo(msqid);
    //메시지를 보낸다.
    if(msgsnd(msqid,&msg,sizeof(struct real_data),0)==-1){
        printf("msgsnd failed\n");
        exit(0);
    }
    printf("message sent\n");
    //메시지 보낸 후 msqid_ds를 한번 보자.
    printMsgInfo(msqid);
}
```



# Message queue

- msq\_receiver1.c

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdlib.h>
#include <string.h>
struct real_data{
    short age;
    char name[16];
};
struct message{
    long msg_type;
    struct real_data data;
};
```

```
int main(){
    key_t key=12345;
    int msqid;
    struct message msg;

    //받아오는 쪽의 msqid얻어오기
    if((msqid=msgget(key,IPC_CREAT|0666))== -1){
        printf("msgget failed\n");
        exit(0);
    }
    //메시지를 받는다.
    if(msgrcv(msqid,&msg,sizeof(struct real_data),0,0)== -1){
        printf("msgrcv failed\n");
        exit(0);
    }

    printf("name : %s, age : %d\n",msg.data.name,msg.data.age);

    //이후 메시지 큐를 지운다.
    if(msgctl(msqid,IPC_RMID,NULL)== -1){
        printf("msgctl failed\n");
        exit(0);
    }
}
```

# Message queue

- msq\_sender2.c & msq\_receiver2.c
  - 키보드 입력을 통해 message queue에 저장할 데이터를 입력 받음
  - receiver에서 데이터 출력



# Message queue

- msq\_sender2.c

```
struct real_data{  
    char age[16];  
    char name[16];  
};  
st  
    msq_sender1.c에  
    서 이 부분 수정  
    a;  
};
```

```
int main(){  
    key_t key=12345;  
    int msqid;  
    char name[16];  
    char age[16];  
  
    struct message msg;  
    msg.msg_type=1;  
    printf("Your name?: ");  
    fgets(name, sizeof(name), stdin);  
    printf("Your age?: ");  
    fgets(age, sizeof(name), stdin);  
  
    strcpy(msg.data.age, age);  
    strcpy(msg.data.name, name);  
  
    //msqid를 얻어옴.  
    if((msqid=msgget(key, IPC_CREAT|0666))==-1){  
        printf("msgget failed\n");  
        exit(0);  
    }  
}
```

msq\_sender1.c에  
서 이 부분 수정

# Message queue

- msq\_receiver2.c

```
struct real_data{  
    char age[16];  
    char name[16];  
};  
st  
};
```

msq\_receiver1.c에  
서 이 부분 수정

```
int main(){  
    key_t key =12345;  
    int msqid;  
    struct message msg;  
  
    if((msqid=msgget(key,IPC_CREAT|0666))== -1){  
        printf("msgget failed\n");  
        exit(0);  
    }  
  
    if(msgrcv(msqid, &msg, sizeof(struct real_data), 0, 0)==-1){  
        printf("msgrcv failed\n");  
        exit(0);  
    }  
  
    printf("name: %s, age: %s n", msg.data.name, msg.data.age);  
  
    if(msgctl(msqid,IPC_RMID,NULL)==-1){  
        printf("msgctl failed\n");  
        exit(0);  
    }  
}
```

msq\_receiver1.c에  
서 이 부분 수정



# Thank you

---

