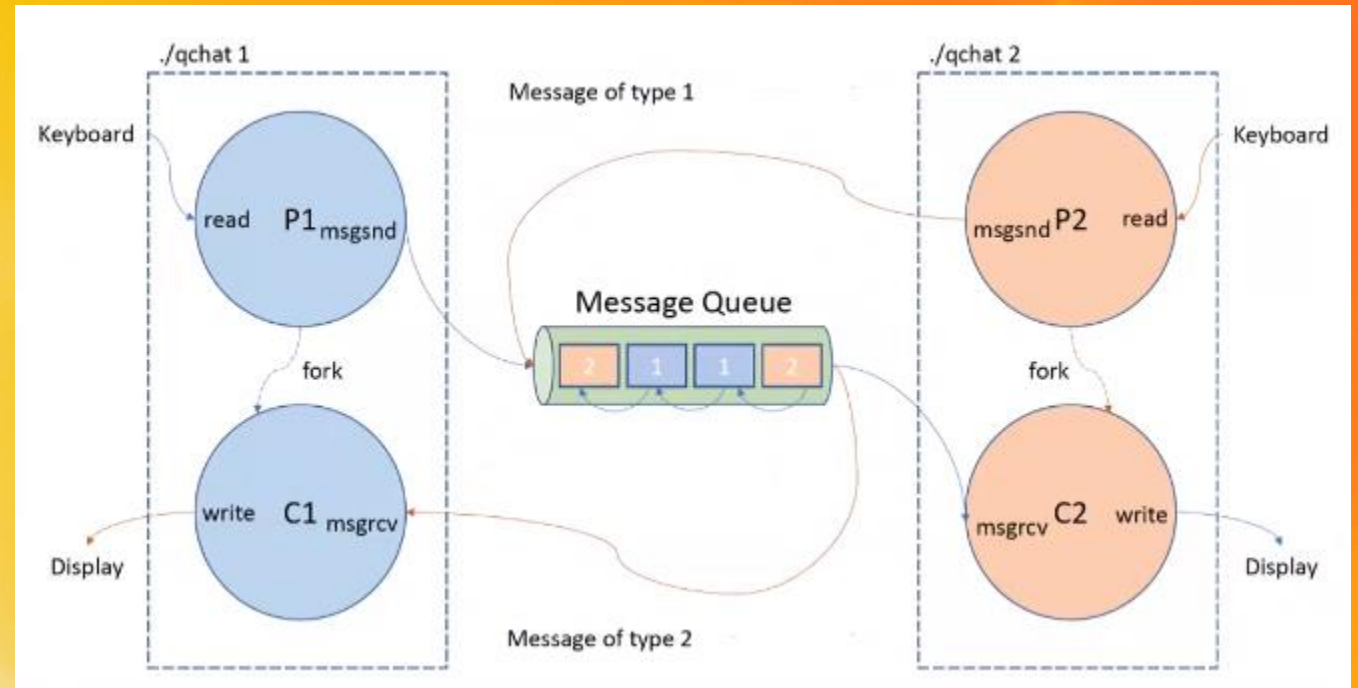


Message Queue



```
static void forceKill()
{
    wait();
    exit(EXIT_SUCCESS);
}
```

จะถูกเรียกโดย signal ใช้ในการแก้ zombie

```
typedef struct a_msg
{
    long int msg_type;
    char data[MAX_BUF];
}a_msg;
```

สร้าง struct เก็บค่า type และ data
เพื่อใช้กับ msgrcv

```
int main(int argc, char*argv[])
{
    a_msg a_msg;
    int msg_id, child, nbytes;

    if(argc!=2)
    {
        fprintf(stderr, "Usage: %s <[ 1 2 ]>\n", *argv);
        exit(EXIT_FAILURE);
    }

    msg_id = msgget((key_t)11235, 0666 | IPC_CREAT);
    if(msg_id==-1)
    {
        fprintf(stderr, "msgget failed");
        exit(EXIT_FAILURE);
    }

    argv++;
    signal(SIGALRM, forceKill);
}
```

ถ้าใส่ argument ไม่ถูก
จะจบโปรแกรม

หา id ของ msg
ที่มี key เป็น 11235
โดยให้ permission read/write
ถ้าไม่มีให้สร้างใหม่

```

if(strcmp(*argv,"1")==0)
{
    child= fork();
    a_msg.msg_type = 1;
    switch(child)
    {
        case -1: perror("Forking Failed");
                exit(EXIT_FAILURE);
        case 0 : while(strncmp(a_msg.data,"end chat",8))
                {
                    if(msgrcv(msg_id, (void *)&a_msg, sizeof(a_msg), 2, 0) == -1)
                    {
                        fprintf(stderr,"msgrcv failed");
                        kill(child,SIGALRM);
                        exit(EXIT_FAILURE);
                    }
                    write(1,"User 2:", 7);
                    write(1, a_msg.data, MAX_BUF);
                }
                kill(child, SIGALRM);
                break;
    }
}

```

เช็คว่าเป็นแพท 1 หรือไม่

สร้าง process ใหม่

ทำจนกว่าจะเจอ"end chat"

รับจาก msg_id ที่ data มี type 2

ถ้ารับค่าที่ใหญ่เกินไป หรือ msg_id ถูกทำลาย

ส่งสัญญาณให้ parent เพื่อออกloop


```

default : while(strncmp(a_msg.data,"end chat",8))
{
    memset(a_msg.data, 0, MAX_BUF);
    nbytes = read(0,a_msg.data,MAX_BUF);
    if (nbytes>1)
    {
        if(msgsnd(msg_id, (void*)&a_msg, sizeof(a_msg), 0) == -1)
        {
            fprintf(stderr,"msgsnd failed");
            kill(child, SIGKILL);
            wait();
            exit(EXIT_FAILURE);
        }
    }
}
kill(child,SIGKILL);
wait();

```

ถ้ามี SIGALRM จะให้ทำforceKill

ถ้ามีค่าให้เขียนลง msg_id
ถ้าเขียนไม่ได้ให้จบการทำงาน

เมื่อออกloopให้ส่ง SIGKILL ฆ่าลูก
แล้วรอจนลูกตายค่อยไปต่อ

```

else if(strcmp(*argv,"2")==0)
{
    child= fork();
    a_msg.msg_type = 2;
    switch(child)
    {
        case -1: perror("Forking Failed");
                exit(EXIT_FAILURE);
        case 0 : while(strncmp(a_msg.data,"end chat",8))
                {
                    if(msgrcv(msg_id, (void *)&a_msg, sizeof(a_msg), 1, 0) == -1)
                    {
                        fprintf(stderr,"msgrcv failed");
                        kill(child,SIGALRM);
                        exit(EXIT_FAILURE);
                    }
                    write(1,"User 1:", 7);
                    write(1, a_msg.data, MAX_BUF);
                }
                kill(child, SIGALRM);
                break;
        default : while(strncmp(a_msg.data,"end chat",8))
                {
                    memset(a_msg.data, 0, MAX_BUF);
                    nbytes = read(0,a_msg.data,MAX_BUF);
                    if (nbytes>1)
                    {
                        if(msgsnd(msg_id, (void*)&a_msg, sizeof(a_msg), 0) == -1)
                        {
                            fprintf(stderr,"msgsnd failed");
                            kill(child, SIGKILL);
                            wait();
                            exit(EXIT_FAILURE);
                        }
                    }
                }
                kill(child, SIGKILL);
                wait();
    }
}

```

ทำงานเหมือนกับแชท 1 แต่

ให้ลูกอ่านจาก type 1

ให้พ่อแม่เขียนโดยเป็น
type 2