CL205 – Operating Systems Lab

Lab#15 – IPC (Message Passing)

Message Passing

Communication takes place by exchange of messages

If two processes P1 & P2 wish to communicate, they need to:

- Establish communication link between them
- Communication link can be uni/bi directional, and associated with a single pair of communicating processes.
- Exchange messages via send(message), receive(message)
- OS Message Queue is a linked list of messages. Queue identified by message queue identifier.

OS Message Queue is a linked list of messages, Queue identified by message queue identifier.

```
struct msg
{
        long mtype;
        char mtext[MSGLENGTH];
};
```

This structure must be included in each process sharing messages.

Type = 0 receives next msg

Type = +ive receives next msg where type matches

Type = -ive receives 1 st msg where type < abs(-ive)

Example
P1 (Message Sender)

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#include <string.h>
#define MSGSZ 128

typedef struct msgbuf
```

```
{
        long mtype;
        char mtext[MSGSZ];
} message_buf;

void main()
{
        int msqid;
        int msgflg = IPC_CREAT | 0666;
        key_t key;
        message_buf sbuf;
        size_t buf_length;
        key = 1234;
        msqid = msgget(key, msgflg );
        sbuf.mtype = 1;
        strcpy(sbuf.mtext, "Did you get this?");
        buf_length = strlen(sbuf.mtext) + 1 ;
        msgsnd(msqid, &sbuf, buf_length, IPC_NOWAIT);
}
```

P2 (Reciever)

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

#define MSGSZ 128

typedef struct msgbuf
{
        long mtype;
        char mtext[MSGSZ];
} message_buf;

void main()
{
        int msqid;
        key_t key;
        message_buf rbuf;
        key = 1234;
        msqid = msgget(key, 0666);
        msgrcv(msqid, &rbuf, MSGSZ, 1, 0);
        printf("%s\n", rbuf.mtext);
}
```