```
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <sys/ipc.h>
#include <fcntl.h>
#include "job.h"

/*
* command syntax
*	enq [-p num] e_file args
*/
void usage()
{
	printf("Usage:	enq [-p num] e_file args\n"
		"\t-p num\t\t specify the job priority\n"
		"\te_file\t\t the absolute path of the exefile\n"
		"\targs\t\t the args passed to the e_file\n");
}

int main(int argc,char *argv[])
{
	int	p = 0;
	int	fd;
	char c, *offset;
	struct jobcmd enqcmd;

	if (argc == 1) {
		usage();
		return 1;
	}

	while (--argc > 0 && (*++argv)[0] == '-') {

		while ((c = *++argv[0]))
			switch (c) {
			case 'p':
				p = atoi(*(++argv));
				argc--;
				break;
			default:
				printf("Illegal option %c\n",c);
				return 1;
			}
	}

	if (p < 0 || p > 3) {
		printf("invalid priority: must between 0 and 3\n");
		return 1;
	}

	enqcmd.type = ENQ;
	enqcmd.defpri = p;
	enqcmd.owner = getuid();
	enqcmd.argnum = argc;
	offset = enqcmd.data;

	while (argc-- > 0) {
```

```
                strcpy(offset,*argv);
                strcat(offset,":");
                offset = offset + strlen(*argv) + 1;
                argv++;
        }

#ifdef DEBUG
        printf("enqcmd cmdtype\t%d\n"
                "enqcmd owner\t%d\n"
                "enqcmd defpri\t%d\n"
                "enqcmd data\t%s\n",
                enqcmd.type, enqcmd.owner, enqcmd.defpri, enqcmd.data);
#endif

        if ((fd = open(FIFO,O_WRONLY)) < 0)
                error_sys("enq open fifo failed");

        if (write(fd,&enqcmd,DATALEN)< 0)
                error_sys("enq write failed");

        close(fd);
        return 0;
}
```