

```

#ifndef JOB_H
#define JOB_H

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdarg.h>
#include <signal.h>
#include <sys/types.h>

#define FIFO "/root/home/SVRFIFO"

#ifndef DEBUG
#define DEBUG
#endif

#undef DEBUG

#define BUFLen 100
#define GLOBALFILE "screendump"
//枚举作业状态：就绪，运行，完成
enum jobstate
{
    READY, RUNNING, DONE
};
//枚举命令类型：进队，出队，当前状态查询
enum cmdtype
{
    ENQ = -1, DEQ = -2, STAT = -3
};

/* this is data passed in fifo */
//作业调度命令结构
struct jobcmd {
    enum cmdtype type;//作业命令类型
    int argnum;//参数类型
    int owner;//作业的 owner
    int defpri;//默认优先级
    char data[BUFLen];//数据类型
};

#define DATALEN sizeof(struct jobcmd)
#define error_sys printf

```

```

struct jobinfo {
    int    jid;                /* 作业 id */
    int    pid;                /* 进程 id */
    char** cmdarg;             /* 执行的命令或者参数 the command & args to execute */
    int    defpri;             /* 默认优先权 default priority */
    int    curpri;             /* 当前的优先权 current priority */
    int    ownerid;            /* 作业拥有者 id the job owner id */
    int    wait_time;          /* 在等待队列中等待的时间 the time job in waitqueue */
    time_t create_time;        /* 创建作业的时间 the time job create */
    int    run_time;           /* 作业运行的时间 the time job running */
    enum    jobstate state;    /* 作业状态 job state */
};

struct waitqueue {            /* 双向的链表 double link list */
    struct waitqueue *next;    //下一个等待作业
    struct jobinfo *job;       //当前等待作业的信息
};

void schedule();//调度函数
void sig_handler(int sig, siginfo_t *info, void *notused);//信号处理
int  allocjid();//分配作业 id
void do_enq(struct jobinfo *newjob, struct jobcmd enqcmd);//入队函数
void do_deq(struct jobcmd deqcmd);//出队函数
void do_stat();//显示作业状态
void updateall();//更新所有作业信息
struct waitqueue* jobselect();//等待队列中选择作业
void jobswitch();//作业转换

#endif

```