```
#ifndef __JOBS_H__
#define __JOBS_H__
#include <pthread.h>

// job structure to hold information about each job being run
// Implementation inspired by work on Lab 12, which also used structures to run a program on
threads

typedef struct _job {
   int jobid; // Number of job in the queue
   char *job_comm; // The command/name of the job
   char *status; // The status of the job e.g., Running, Waiting
   char *outFile; // The file that stdout is redirected to <jobid>.out
   char *errFile; // The file that stderr is redirected to <jobid>.err
   pthread_t jtid; // Associate job with a thread to run it
} job;

// From the queue.h

typedef struct _queue {
        int size;    /* maximum size of the queue */
        job **item; /* queue buffer */ // Job array
        int start;   /* index to the start of the queue */
        int end;     /* index to the end of the queue */
        int count;   /* no. of elements in the queue */
} queue;

queue *queue_init(int n);
int queue_insert(queue *q, job *item);
job *queue_delete(queue *q);
// queue display removed due to having no purpose here
void queue_destroy(queue *q);

// Job commands
job init_job(int id, char *command);
void show_jobs(int arrlen, job *jobList);

#endif
```