

```

1  #ifndef __LINKED_LIST_H__
2  #define __LINKED_LIST_H__
3
4  #include <stdlib.h>
5
6  typedef struct node {
7      void *job;
8      size_t job_size;
9      void *owner;
10     struct node *next;
11     struct node *prev;
12 } list;
13
14 list *create_list () {
15     list *new_list = (list *) malloc(sizeof(list));
16     new_list->job = NULL;
17     new_list->job_size = 0;
18     new_list->owner = NULL;
19     new_list->next = NULL;
20     new_list->prev = NULL;
21
22     return new_list;
23 }
24
25 void add_job (list *l, void *job, size_t job_size, void *owner) {
26     struct node *job_node = (struct node *) malloc(sizeof(struct node));
27     job_node->job = job;
28     job_node->job_size = job_size;
29     job_node->owner = owner;
30
31     struct node *current = l;
32     while (current->next != NULL) {
33         if (job_size < current->job_size) {
34             job_node->next = current;
35             job_node->prev = current->prev;
36             current->prev->next = job_node;
37             current->prev = job_node;
38
39             return;
40         }
41
42         current = current->next;
43     }
44
45     current->next = job_node;
46     job_node->prev = current;
47 }
48
49
50
51 void *get_job (list *l) {
52     list *nd = l->next;
53     if (!nd) return NULL;
54     void *job = nd->job;
55
56     if (l->next->next)
57         l->next->next->prev = l;
58     l->next = l->next->next;
59
60     return nd;
61 }
62
63 #endif
64

```