```c
#ifndef __LINKED_LIST_H__
#define __LINKED_LIST_H__

#include <stdlib.h>

typedef struct node {
    void *job;
    size_t job_size;
    void *owner;
    struct node *next;
    struct node *prev;
} list;

list *create_list () {
    list *new_list = (list *) malloc(sizeof(list));
    new_list->job = NULL;
    new_list->job_size = 0;
    new_list->owner = NULL;
    new_list->next = NULL;
    new_list->prev = NULL;

    return new_list;
}

void add_job (list *l, void *job, size_t job_size, void *owner) {
    struct node *job_node = (struct node *) malloc(sizeof(struct node));
    job_node->job = job;
    job_node->job_size = job_size;
    job_node->owner = owner;

    struct node *current = l;
    while (current->next != NULL) {
        if (job_size < current->job_size) {
            job_node->next = current;
            job_node->prev = current->prev;
            current->prev->next = job_node;
            current->prev = job_node;

            return;
        }

        current = current->next;
    }

    current->next = job_node;
    job_node->prev = current;
}



void *get_job (list *l) {
    list *nd = l->next;
    if (!nd) return NULL;
    void *job = nd->job;

    if (l->next->next)
        l->next->next->prev = l;
    l->next = l->next->next;

    return nd;
}

#endif

```