

Факультет программной инженерии и компьютерной техники

Низкоуровневое программирование

Лабораторная работа №5

Преподаватель:

Логинов Иван Павлович

Выполнил:

Рождественский Никита Сергеевич

Р33113

## Задание:

* Реализовать все функции, необходимые для работы со связным списком, из [задания на Stepik](https://stepik.org/lesson/408350/step/8). Необходимо переиспользовать код из [предыдущего задания](https://gitlab.se.ifmo.ru/low-level-programming/assignment-4-linked-list).
* Реализуйте все функции из [задания на Stepik](https://stepik.org/lesson/408351/step/6).
* Также реализуйте функции:

bool save(const struct list\* lst, const char\* filename);

bool load(struct list\*\* lst, const char\* filename);

bool serialize(const struct list\* lst, const char\* filename);

bool deserialize(struct list\*\* lst, const char\* filename);

Выполнение:

int64\_t check\_file(FILE\* file) {

if (errno || ferror(file)) {

fclose(file);

return 0;

} else return 1;

}

int64\_t close\_file(FILE\* file) {

fclose(file);

if (errno) return 0;

else return 1;

}

FILE\* open\_file(const char\* filename, const char\* mode) {

errno = 0;

FILE\* file = fopen(filename, mode);

if (errno) return NULL;

else return file;

}

bool save(const struct list\* lst, const char\* filename){

FILE\* file=open\_file(filename,"w");

if(file==NULL) return 0;

struct list\* list=node\_create(lst->value);

list->next=lst->next;

while (list->next){

fprintf(file,"% " PRId64,list->value);

list=list->next;

}

fprintf(file,"% " PRId64,list->value);

return fclose(file);

}

bool load(struct list\*\* lst, const char\* filename){

FILE\* file=open\_file(filename,"r");

if(file==NULL) return 0;

int64\_t val;

while(true){

fscanf(file,"%" SCNd64,&val);

list\_add\_front(lst,val);

if (feof(file)) break;

if (!check\_file(file)) return 0;

}

\*lst=list\_reverse(\*lst);

return fclose(file);

}

bool serialize(const struct list\* lst, const char\* filename){

FILE\* file = open\_file(filename, "w");

if (file == NULL) return 0;

struct list\* list=node\_create(lst->value);

list->next=lst->next;

while (list->next){

fwrite(&list->value,sizeof(list->value),1,file);

if (!check\_file(file)) return 0;

list=list->next;

}

fwrite(&list->value,sizeof(list->value),1,file);

return close\_file(file);

}

bool deserialize(struct list\*\* lst, const char\* filename){

FILE\* file=open\_file(filename,"r");

if(file==NULL) return 0;

int64\_t val;

while (true){

fread(&val, sizeof(int64\_t), 1, file);

if (feof(file)) break;

if (!check\_file(file)) return 0;

list\_add\_front(lst,val);

}

\*lst=list\_reverse(\*lst);

return fclose(file);

}

struct list\* list\_read() {

int64\_t val;

struct list\* new\_list = NULL;

puts("Enter list:");

while(scanf("%" SCNd64, &val) == 1) {

list\_add\_front( &new\_list,val);

}

new\_list=list\_reverse(new\_list);

return new\_list;

}

Вывод: В ходе выполнения данной лабораторной работы были реализованы функции высшего порядка для работы со списком и работа с файловой системой на языке C.