

25. Журналирование.

План ответа:

- 1 Назначение
- 2 идеи реализации

1 Журналирование – процесс записи информации о происходящих с каким-то объектом (или в рамках какого-то процесса) событиях в журнал (например, в файл). Этот процесс часто называют также аудитом.

2 Идеи реализации

Файловую переменную для журнала определяют глобальной и объявляют во всех файлах реализации проекта.

Это позволяет вызывать функции записи в файл для журналирования отовсюду.

```
// log.c
#include <stdio.h>
FILE *flog;
int log_init(const char* name)
{
    flog = fopen(name, "w");
    if(!flog)
        return 1;
    return 0;
}
void log_close(void)
{
    fclose(flog);
}

// log.h
#ifndef __LOG_H__
#define __LOG_H__
extern FILE *flog;
int log_init(const char*name);
void log_close(void);
#endif // __LOG_H__
```

Для записи в журнал можно написать собственные функции, где переменная-указатель на файл с журналом может быть как глобальной, так и статической переменной

```
// log.c
#include <stdio.h>
static FILE *flog;
int log_init(const char* name)
{
    flog = fopen(name, "w");
    if(!flog)
        return 1;
    return 0;
}
FILE* log_get(void)
{
    return flog;
}
void log_close(void)
{
    fclose(flog);
}

// log.h
#ifndef __LOG_H__
#define __LOG_H__
#include <stdio.h>
int log_init(const char* name);
FILE* log_get(void);
void log_close(void);
#endif // __LOG_H__
```

Реализация с помощью функций с переменным числом параметров

```

// log.c
#include <stdio.h>
#include <stdarg.h>
static FILE* flog;
int log_init(const char*name)
{
    flog = fopen(name, "w");
    if(!flog)
        return 1;
    return 0;
}
void log_message(const char* format, ...)
{
    va_list args;
    va_start(args, format);
    vfprintf(flog, format, args);
    va_end(args);
}
void log_close(void)
{
    fclose(flog);
}

// log.h
#ifndef __LOG__H__
#define __LOG__H__
#include <stdio.h>
int log_init(const char* name);
void log_message(const char* format, ...);
void log_close(void);
#endif // __LOG__H__

```