

Анализ клиента:

При запуске проверяет наличие сервера в системе. Проверка выполняется отправкой сигнала SIGWINCH всем процессам, которым клиент может отправлять сигнал. Если сервера нет, то клиент выходит.

```
#include <unistd.h>

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <signal.h>

#define SLEEP_TIME 935000000L // 935 mili seconds
int server_pid=0;
int server_is_ready=1;

#define error_in_client_ "/tmp/error_in_client_"
#define REPORT_ERROR(a) \
{ \
    FILE *f=fopen(error_in_client_,"w+"); \
    fprintf(f," id_sender  %d  |  %s\n",getpid(),a); \
    fclose(f); \
    _exit(0);\
} \

// сигнал
void signal_synchronization(int sig_num,siginfo_t *siginfo, void *ucontext)
{
    server_pid=siginfo->si_pid;
}

void signal_work_is_done(int sig_num,siginfo_t *siginfo, void *ucontext)
{
    server_is_ready=1;
}

int becomeDaemon(struct timespec *req,struct timespec *rem)
{
    switch (fork())
    {
        case -1: return -1; //error
        case 0: break; //child
        default: _exit(0); //parent
    }

    if(chdir("/")==-1)
    {
        printf("cannot change directory");
        return 0;
    };

    struct sigaction neu;
    neu.sa_sigaction=signal_synchronization;
    neu.sa_flags=SA_SIGINFO;
```

```

if(sigaction(SIGINT,&neu,NULL)==-1)
    REPORT_ERROR("cannot establish\n");

if( kill(-1,SIGWINCH)==-1) // отправили сигнал, ждем ответа
    // в ответ сервер должен отправить сигнал SIGINT
    // если ответа не будет, то глобальная переменная
    // server_pid останется ==0, что говорит об отсутствии
    //сервера
    REPORT_ERROR("cannot send synchronization signal");

    sleep(2);
//has elapsed enough time

if(server_pid==0)
    REPORT_ERROR("server not found");

    union sigval v;
    v.sival_int=1; // отправляем идентификационный набор чисел 1,2,3,4,5

if(sigqueue(server_pid,SIGRTMIN,v)==-1)
    REPORT_ERROR("cannot send synchronization signal 1st");

v.sival_int++;

if(sigqueue(server_pid,SIGRTMIN,v)==-1)
    REPORT_ERROR("cannot send synchronization signal 1st");
v.sival_int++;

if(sigqueue(server_pid,SIGRTMIN,v)==-1)
    REPORT_ERROR("cannot send synchronization signal 1st");
v.sival_int++;
if(sigqueue(server_pid,SIGRTMIN,v)==-1)
    REPORT_ERROR("cannot send synchronization signal 1st");
v.sival_int++;
if(sigqueue(server_pid,SIGRTMIN,v)==-1)
    REPORT_ERROR("cannot send synchronization signal 1st");
v.sival_int++;

sleep(1);

struct sigaction work_done;

work_done.sa_sigaction=signal_work_is_done;
work_done.sa_flags=SA_SIGINFO;

if(sigaction(SIGALRM,&work_done,NULL)==-1)
    REPORT_ERROR("cannot establish\n");

rem->tv_nsec=SLEEP_TIME;

int prev_mask=0;

while(1)
{
    prev_mask=sigsetmask(0xFFFFfff);
    if(nanosleep(rem,req)==-1)

```

```
        exit(0);
sigsetmask(prev_mask);

    if(server_is_ready)
    {   if( kill(server_pid,SIGALRM)==-1)
        exit(0);
        server_is_ready=0;
    }
    else
    if(kill(server_pid,0)==-1)
    exit(0);
}

}

main()
{
    struct timespec *req;
    struct timespec *rem;

    req=new timespec;
    rem=new timespec;
    req->tv_sec=0;
    req->tv_nsec=0;

    becomeDaemon(req, rem);

    delete req;
    delete rem;

}
```