```c
/* https://sploitfun.wordpress.com/2015/02/10/understanding-glibc-malloc/ */

/* Per thread arena example. */
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
#include <sys/types.h>

void* threadFunc(void* arg) {
    printf("Before malloc in thread 1\n");
    getchar();
    char* addr = (char*) malloc(1000);
    printf("After malloc and before free in thread 1\n");
    getchar();
    free(addr);
    printf("After free in thread 1\n");
    getchar();
}

int main() {
    pthread_t t1;
    void* s;
    int ret;
    char* addr;

    printf("Welcome to per thread arena example::%d\n",getpid());
    printf("Before malloc in main thread\n");
    getchar();
    addr = (char*) malloc(1000);
    printf("After malloc and before free in main thread\n");
    getchar();
    free(addr);
    printf("After free in main thread\n");
    getchar();
    ret = pthread_create(&t1, NULL, threadFunc, NULL);
    if(ret)
        {
            printf("Thread creation error\n");
            return -1;
        }
    ret = pthread_join(t1, &s);
    if(ret)
        {
            printf("Thread join error\n");
            return -1;
        }
    return 0;
}
```