

OS Lab5

```
PTHREAD_CLEANUP_PUSH(3)    Linux Programmer's Manual    PTHREAD_CLEANUP_PUSH(3)

NAME
    pthread_cleanup_push, pthread_cleanup_pop - push and pop thread cancel-
    lation clean-up handlers

SYNOPSIS
    #include <pthread.h>

    void pthread_cleanup_push(void (*routine)(void *),
                               void *arg);
    void pthread_cleanup_pop(int execute);

    Compile and link with -pthread.
```

Поток может назначить некоторую функцию для вызова в момент его завершения примерно так же, как это делается для процессов с помощью функции `atexit`, которая регистрирует функции, запускаемые при завершении процесса. Эти функции называют функциями обработки выхода из потока. Поток может зарегистрировать несколько таких функций обработки выхода. Обработчики заносятся в стек — это означает, что они будут вызываться в порядке, обратном порядку их регистрации. Функция `pthread_cleanup_push` регистрирует функцию `rt`, которая будет вызвана с аргументом `arg`, когда поток выполнит одно из следующих действий: - вызовет функцию `pthread_exit`; - ответит на запрос о принудительном завершении; - вызовет функцию `pthread_cleanup_pop` с ненулевым аргументом `execute`

Если аргумент `execute` имеет значение 0, функция обработки выхода из потока вызываться не будет. В любом случае функция `pthread_cleanup_pop` удаляет функцию-обработчик, зарегистрированную последним обращением к функции `pthread_cleanup_push`.

Ограничение, связанное с этими функциями, заключается в том, что они могут быть реализованы в виде макроопределений, и тогда они должны использоваться в паре, в пределах одной области видимости в потоке. Макроопределение функции `pthread_cleanup_push` может включать символ `{`, и тогда парная ей скобка `}` будет находиться в макроопределении `pthread_cleanup_pop`.

Функции-обработчики вызываются в стековом порядке (Last In First Out), т.е. в порядке, обратном тому, в котором они устанавливались, и из того контекста, в котором они устанавливались. Иными словами, в качестве параметров функций-обработчиков можно использовать указатели на локальные переменные, определенные в тех блоках, в которых стоял вызов соответствующего `pthread_cleanup_push(3C)`.

Каждый вызов `pthread_cleanup_push(3C)` должен сопровождаться соответствующим вызовом парной макрокоманды `pthread_cleanup_pop(3C)`. Эта макрокоманда должна вызываться в пределах того же блока, в котором была вызвана соответствующая `pthread_cleanup_push(3C)`.

Нарушение этого требования приведет к ошибке компиляции. В действительности, макрокоманды `pthread_cleanup_push/pop` содержат, соответственно, открывающую и закрывающую фигурную скобки. Это не является требованием стандарта POSIX, но поможет