

## OS Lab 4

### Как прервать нить?

Один поток может передать запрос на принудительное завершение другого потока в том же процессе, обратившись к функции `pthread_cancel`. По умолчанию вызов `pthread_cancel` заставляет указанный поток вести себя так, словно он вызвал функцию `pthread_exit` с аргументом `PTHREAD_CANCELED`. Однако поток может отвергнуть запрос или как-то иначе отреагировать на него. Обратите внимание, что функция `pthread_cancel` не ждет завершения потока. Она просто посылает запрос. В момент запуска потока значение его атрибута `cancelability state` устанавливается равным `PTHREAD_CANCEL_ENABLE`. Если поток установит значение этого атрибута равным `PTHREAD_CANCEL_DISABLE`, вызов функции `pthread_cancel` не будет приводить к завершению потока. Вместо этого запрос на принудительное завершение встает в режим ожидания. Когда поток опять разрешит возможность принудительного завершения, он откликнется на ожидающий запрос в ближайшей точке выхода

#### SYNOPSIS

```
#include <pthread.h>

int pthread_cancel(pthread_t thread);

Compile and link with -pthread.
```

#### ERRORS

```
ESRCH  No thread with the ID thread could be found.
```

Cancel state (состояние прерывания) определяет, разрешено ли прерывание нити как таковое. Т.е. этот атрибут может иметь два значения – разрешено или запрещено. Если прерывание разрешено, нить немедленно получает сообщение о попытке ее прервать (хотя, в зависимости от cancel type, может отреагировать на это

сообщение лишь через некоторое время). Если прерывание запрещено, попытки прерывания нити накапливаются. После того, как прерывания все-таки разрешат, нить получит сигналы о накопившихся попытках.

Переключение состояния прерывания осуществляется функцией `pthread_setcancelstate(3C)`.

```
SYNOPSIS      Cancel state (состояние прерывания) определяет, разрешено
#include <pthread.h> ние нити как таковое. Т.е. этот атрибут может иметь
                значения – разрешено или запрещено. Если прерывание разреш
int pthread_setcancelstate(int state, int *oldstate);
int pthread_setcanceltype(int type, int *oldtype);
                зависимости от cancel type, может отреагировать на
Compile and link with -pthread.
```

Первый параметр этой функции входной и может принимать значения `PTHREAD_CANCEL_ENABLE` (прерывание разрешено) и `PTHREAD_CANCEL_DISABLE` (прерывание запрещено). Эти значения – препроцессорные макроопределения, содержащиеся в файле `pthread.h`. Вызов функции с другими значениями первого параметра приведет к ошибке `EINVAL`. Второй параметр функции – выходной, содержит указатель на переменную, в которой будет размещено старое значение типа прерывания. В качестве этого указателя можно передать `NULL`, в этом случае старое значение состояния будет потеряно. По умолчанию, нить создается с разрешенными прерываниями.

Cancel type (тип прерывания) определяет, в какие моменты нить проверяет сообщения о прерываниях. Этот атрибут может принимать два значения – `PTHREAD_CANCEL_DEFERRED` (отложенное прерывание) и `PTHREAD_CANCEL_ASYNCCHRONOUS` (асинхронное прерывание). По умолчанию, нить создается с отложенным типом прерываний. Что означает каждое из возможных значений этого атрибута, описывается далее в этом разделе.

Установка типа прерывания осуществляется функцией `pthread_setcanceltype(3C)`. Схема передачи параметров этой функции аналогична `pthread_setcancelstate(3C)`.

Тип и состояние прерывания могут быть заданы в момент создания нити при помощи установки соответствующих полей в структуре `pthread_attr_t`. Обратите внимание, что оба эти атрибута задаются либо в момент создания нити, либо самой нитью. Внешними по отношению к нити средствами их изменить невозможно. Поэтому проверка значений этих атрибутов в определенные моменты времени не может приводить к ошибкам соревнования.

Асинхронное прерывание означает, что библиотека прерывает нить как можно скорее (хотя во многих ситуациях не удастся гарантировать, чтобы это происходило точно в тот момент, когда другая нить вызвала `pthread_cancel(3C)`). Асинхронное прерывание требует тщательного анализа всех возможных моментов, когда оно может произойти, и обработки всех ситуаций, связанных с прерываниями в неудачные моменты. Так, если прерывание произойдет во время работы с библиотекой, которая не считается *thread-safe*, внутренние данные этой библиотеки могут остаться в несогласованном состоянии.

Отложенное прерывание означает, что нить получает сообщение о прерывании лишь в определенные моменты, известные как точки прерывания (cancellation point). Эти точки, в свою очередь, делятся на две категории – явные и неявные. Явные точки прерывания – это вызовы функции `pthread_testcancel(3C)`. Неявные точки