

Лабораторная 5

Использованные функции :

Сигнатура функции :

```
#include <pthread.h>

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

Эти функции управляют стеком обработчиков очистки отмены потока, который вызвал эти функции.

Функция `pthread_cleanup_push()` помещает функцию на вершину стека обработчиков очистки. При последующем вызове этой функции в качестве аргумента будет указан аргумент `arg`.

Функция `pthread_cleanup_pop()` удаляет процедуру в верхней части стека обработчиков очистки и выполняет ее, если значение, переданное как аргумент не равно нулю.

Обработчик очистки - это функция , которая автоматически выполняется при отмене потока (или при других обстоятельствах, описанных ниже); он может, например, разблокировать мьютекс, чтобы он стал доступен другим потокам в процессе.

Обработчик очистки отмены извлекается из стека и выполняется при следующих обстоятельствах:

1. Когда поток отменяется, все обработчики очистки извлекаются и выполняются в порядке, обратном порядку , в котором они были помещены в стек.
2. Когда поток завершается вызовом `pthread_exit(3)`, все обработчики очистки выполняются, как описано в предыдущем пункте. (Обработчики очистки не вызываются, если поток завершается выполнением `return` из функции потока.)
3. Когда поток вызывает `pthread_cleanup_pop()` с ненулевым аргументом, извлекается и выполняется самый верхний обработчик очистки

Ограничение, связанное с этими функциями, заключается в том, что они могут быть реализованы в виде макроопределений, и поэтому 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`.

В некоторых операционных системах, например, Mac OS X завершение потока выполнением `return` из функции потока приводит к аварийному завершению.

Проблема в том, что в этих системах функция `pthread_cleanup_push` реализована как макрос, сохраняющий некоторые контекстные данные на стеке. Когда поток 1 возвращает управление между вызовами

`pthread_cleanup_push` и `pthread_cleanup_pop`, содержимое стека затирается и программа в этих системах использует уже поврежденный к этому моменту контекст, когда пытается вызвать обработчики выхода из потока.

Поведение системы при возврате(вызове `return`) между соответствующими парными вызовами `pthread_cleanup_push` и `pthread_cleanup_pop` не регламентируется стандартом Single UNIX Specification и поэтому единственный переносимый способ выполнить возврат между вызовами этих двух функций — обратиться к функции `pthread_exit`.

Возвращаемое значение :

ничего не возвращает

Возможные ошибки :

нет ошибок