



OS2 Lab5

checkbox	<input checked="" type="checkbox"/>
due date	@September 17, 2021
class	OS
Status	approved

Task

Модифицируйте программу упр. 4 так, чтобы дочерняя нить перед завершением распечатывала сообщение об этом. Используйте pthread_cleanup_push.

Notes

```
#include <pthread.h>
void pthread_cleanup_push(void (*rtn)(void *), void *arg);
void pthread_cleanup_pop(int execute);

#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <stdbool.h>
#include <unistd.h>

#define EXECUTE 1

void onExit(void *args){
    printf("terminating thread #%lu\n", pthread_self());
}

void* printString(void *args){
    pthread_cleanup_push(onExit, NULL);
    while (true){
        printf("botay!\n");
    }
    pthread_cleanup_pop(EXECUTE);
    pthread_exit(0);
}

int main() {

    pthread_t endlessThread;

    if (pthread_create(&endlessThread, NULL, printString, NULL) != 0){
        perror("failed to create new thread");
        exit(-1);
    }

    sleep(2);
    printf("\n");

    if (pthread_cancel(endlessThread) != 0){
        perror("failed to cancel execution of new thread");
        exit(-1);
    }

    /*обратите внимание что необходимо заджойнить поток, так как иначе он не будет успевать исполнить функцию onExit(), поскольку функция pthread_cancel не ждет завершения потока. Она просто посылает запрос.*/

    if (pthread_join(endlessThread, NULL) != 0){
        perror("failed to join new thread");
    }
}
```

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 называются 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 `rtn`, 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`.

```
        exit(-1);
    }
    return 0;
}
```

The effect of the use of **return**, **break**, **continue**, and **goto** to prematurely leave a code block described by a pair of **pthread_cleanup_push()** and **pthread_cleanup_pop()** function calls is undefined.

Using **longjmp()** or **siglongjmp()** to jump into or out of a push/pop pair can cause either the matching push or the matching pop statement not getting executed.

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

37

Д.В. Иртегов

Многопоточное программирование с использованием POSIX Threads

понять, к каким именно ошибкам компиляции может привести их неправильное использование.

Reading list

