

```

/*
 * mykill.c : 簡易 kill コマンド
 * (書式: mykill PID...)
 */

#include <stdio.h>
#include <stdlib.h> // strtol のため
#include <signal.h> // kill() のため

/*
 * 以下のプログラムでは、より柔軟にエラーメッセージを作るために
 * perror() ではなく strerror() を使用している。
 *
 * char *strerror(int errnum); エラー番号に対応するエラーメッセージを返す。
 */
#include <string.h> // strerror() のため
extern int errno;

void usage(char *myname) {
    fprintf(stderr, "Usage: %s <PID>...\n", myname);
    exit(1);
}

int main(int argc, char *argv[]) {
    int stat = 0;
    if (argc < 2) usage(argv[0]);

    for (int i=1; i<argc; i++) {
        char *endp;
        int pid = strtol(argv[i], &endp, 10);

        if (*argv[i]!='\0' || *endp!='\0') {
            fprintf(stderr, "%s : %s : arguments must be PID\n", argv[0], argv[i]);
            stat = 1;
        } else if (kill(pid, SIGTERM) < 0) {
            fprintf(stderr, "%s : %d : %s\n", argv[0], pid, strerror(errno));
            stat = 1;
        }
    }

    return stat;
}

/* 実行例
$ xclock &                                <-- 何かプロセスを起動する
[2] 7794
$ mykill 7794                              <-- mykill のテスト
$
[2]+  Terminated: 15                      xclock    <-- bashが終了したJOBをレポート
$ xclock &
[2] 7796
$ xclock &
[3] 7797
$ mykill 7796 7797                        <-- mykill のテスト(複数プロセス)
$
[2]-  Terminated: 15                      xclock    <-- bashが終了したJOBをレポート
[3]+  Terminated: 15                      xclock    <-- bashが終了したJOBをレポート
$ mykill 7796 7797                        <-- mykill のテスト(該当複数なし)
mykill : 7796 : No such process
mykill : 7797 : No such process
$ mykill abc                              <-- mykill のテスト(PID書式エラー)
mykill : abc : arguments must be PID
$
*/

```