

Subject: Lecture on Operating System Lecture Exercise 7.4

From: IKENOYA Katsutoshi <j05002@ie.u-ryukyu.ac.jp>

Date: Mon, 22 Jan 2007 17:45:14 +0900

To: Shinji KONO <kono@ie.u-ryukyu.ac.jp>

学籍番号 : 055702B

問題7.4

kbhit() を Unix で作ってみよう。ダメな例を実装してみて、どれくらいダメか調べよう。

・ソース

ソースの場所 : /home/y05/j05002/OS-Lecture7/mondai7-4/kbhit.c

```
#include <unistd.h>
#include <sys/ioctl.h>

static struct termios tio, tio_new;
static int gc;

int main();
int init_keyboard();
int close_keyboard();
//int kbhit();
int kbhit2(); // ダメな例
int getch();

int main()
{
    int ch = 0;

    init_keyboard();

    while(ch != 'q') {
        sleep(1);
        gc = 0;

        /* キー入力があった時 */
        if(kbhit2()) {
            if(!gc)
                ch = getch();
            else
                ch = gc;

            /* 端末に表示可能な文字であるなら */
            if(isascii(ch) && isprint(ch))
                printf("you hit : %c\n", ch);
            else /* Ctl+D, Ctl+C 等 */
                printf("you hit : [%02x]\n", ch);
        } else {
            /* キー入力がない時 */
            printf(".");
            fflush(stdout);
        }

        close_keyboard();
    }
}
```

省略

```

int kbhit()
{
char ch;
int nread, n;

tio_new.c_cc[VMIN]=0;
// キー入力があったか調べる
nread = ioctl(0, FIONREAD, &n);
tio_new.c_cc[VMIN]=1;
tcsetattr(0, TCSANOW, &tio_new);

// キー入力があれば
if(nread != -1) {
return n; /* n:読み込んだデータバイト数 */
}
return 0;
}

/* ダメな例 */
int kbhit2()
{
{
if( (gc = getchar()) != EOF) {
return gc;
} else {
return -1;
}
}
}

/* キー入力の読み込み */
int getch()
{
{
char ch;
read(0, &ch, 1);
return ch;
}
}

```

・実行結果

(1)kbhit()で実行した場合

```

[j05002@mondai7-4]% ./kbhit
...you hit : a
....you hit : s
.....you hit : c
you hit : d
...you hit : q

```

キー入力がない場合は"."が出力されるのでキー入力待ちの状態にならない。

(2)ダメな場合(kbhit2()で実行)

```

[j05002@mondai7-4]% ./kbhit
a
you hit : a
you hit : [0a]
d
you hit : d
you hit : [0a]
q
you hit : q

```

(1)の場合と違ってキー入力が無い場合は、何も表示されないのでキー入力待ちの状態になっており、CPU時間を無駄に使う。