

스톱워치

8_3_1

소프트웨어학과
2018843103
정현준

8_3_1

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <conio.h>

void time_pass(time_t start);
void display_time(long nhour, long nmin, long nsecond);
void control_watch(void);

int main(void)
{
    display_time(0, 0, 0);
    getch();
    control_watch();
    return 0;
}

void control_watch(void)
{
    char key;
    time_t start;
    start=time(NULL);
    time_pass(start);
    do
    {
        key=getch();
        switch(key)
        {
            case 10 :
                time_pass(start);
            case 32 :
                start=time(NULL);
                time_pass(start);
                break;
```

```
            case 27 :
                exit(0);
                break;
        }
    }while(key!=27);

}

void time_pass(time_t start)
{
    double hour, min, second;
    long nhour, nmin, nsecond;
    time_t now;
    while(!kbhit())

    {
        now=time(NULL);
        second=difftime(now, start);
        hour= (second/3600.);
        nhour= (long) hour;
        min= fmod(second, 3600.)/60;
        nmin= (long) min;
        second= fmod(second, 60);
        nsecond= (long) second;
        display_time(nhour, nmin, nsecond);
    }
}

void display_time(long nhour, long nmin, long nsecond)
{
    system("cls");
    printf("stopwatch\n\n");
    printf("%.2ld시 %.2ld분 %.2ld초 \n\n", nhour, nmin, nsecond);
    printf("Enter:(시작/정지), SPACE:재시작, Esc:종료\n");

}
```

코드 요약

이 프로그램은 간단한 **스톱워치**입니다.

0시 0분 0초에서 시작합니다.

사용자는 Enter 키로 타이머를 시작과 정지하고 Space 키로 리셋하며 Esc 키로 종료할 수 있습니다.

화면은 지속적으로 갱신되어 경과 시간을 표시합니다.

헤더파일

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <conio.h>
```

- `stdio.h`: 기본적인 입력 및 출력 작업을 위한 헤더.
- `stdlib.h`: 메모리 할당, 프로세스 제어 등과 관련된 함수를 포함.
- `time.h`: 시간 및 날짜 관련 함수.
- `math.h`: 수학 관련 함수

헤더파일2

- `conio.h` : `getch()` 및 `kbhit()` 과 같은 콘솔 입출력 함수.

`kbhit()` : 키보드 버퍼에 입력이 있는지 확인하는 함수

키보드 버퍼에 입력이 있으면 0이 아닌 값을 반환하고, 그렇지 않으면 0을 반환합니다.

`getch()` : 키보드로부터 한 문자를 읽는 함수

호출되면 실행이 멈추고 사용자의 키 입력을 기다립니다.

함수 선언

```
void time_pass(time_t start);  
void display_time(long nhour, long nmin, long nsecond);  
void control_watch(void);
```

세 가지 함수가 선언되어 있습니다.
이후에 정의됨.

스톱워치 제어

```
void control_watch(void)
```

스톱워치의 제어를 담당하는 함수입니다. 키 입력에 따라 동작합니다:

- `Enter (10)` : 스톱워치 시작 / 정지.
- `Space (32)` : 스톱워치를 초기화하고 다시 시작.
- `Esc (27)` : 프로그램 종료.

시간 경과

```
void time_pass(time_t start)
```

`start` 시간부터 경과한 시간을 계산하는 함수입니다. 시, 분, 초를 계산하고, 그것을 표시하기 위해 `display_time` 함수를 사용합니다. 키를 누를 때까지 화면을 계속 업데이트합니다.

시간 표시

```
void display_time(long nhour, long nmin, long nsecond)
```

콘솔 화면을 지우고 현재 스톱워치의 시간을 출력하며 사용자에게 스톱워치 제어 방법도 함께 표시합니다.

```
stopwatch
00시 00분 00초
Enter:(시작/정지), SPACE:재시작, Esc:종료
|
```

첫 화면

```
stopwatch
00시 00분 08초
Enter:(시작/정지), SPACE:재시작, Esc:종료
|
```

실행 후

```
stopwatch
```

```
00시 00분 08초
```

```
Enter:(시작/정지), SPACE:재시작, Esc:종료
```

```
|
```

정지 장면

```
stopwatch
```

```
00시 00분 08초
```

```
Enter:(시작/정지), SPACE:재시작, Esc:종료
```

```
-----
```

```
Process exited after 37.17 seconds with return value 0
```

```
계속하려면 아무 키나 누르십시오 . . . |
```

종료 후

macOs에서 실행할 수 있는 코드

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <unistd.h>
#include <termios.h>
#include <fcntl.h>

void time_pass(time_t start);
void display_time(long nhour, long nmin, long nsecond);
void control_watch(void);
int kbhit(void);

int main(void)
{
    display_time(0, 0, 0);
    getchar();
    control_watch();
    return 0;
}

void control_watch(void)
{
    char key;
    time_t start;
    start = time(NULL);
    time_pass(start);
}

while (1)
{
    if (kbhit())
    {
        key = getchar();

        switch (key)
        {
            case '\n': // 엔터 키
                return; // 함수 종료로 정지
            case ' ': // 스페이스바
                start = time(NULL);
                time_pass(start);
                break;
            case 27: // ESC 키
                exit(0);
                break;
        }
    }
}

int kbhit(void)
{
    struct termios oldt, newt;
    int ch;
    int oldf;

    tcgetattr(STDIN_FILENO, &oldt);
    newt = oldt;
    newt.c_lflag &= ~(ICANON | ECHO);
    tcsetattr(STDIN_FILENO, TCSANOW, &newt);
    oldf = fcntl(STDIN_FILENO, F_GETFL, 0);
    fcntl(STDIN_FILENO, F_SETFL, oldf | O_NONBLOCK);

    ch = getchar();

    tcsetattr(STDIN_FILENO, TCSANOW, &oldt);
    fcntl(STDIN_FILENO, F_SETFL, oldf);

    if (ch != EOF)
    {
        ungetc(ch, stdin);
        return 1;
    }

    return 0;
}

void time_pass(time_t start)
{
    double hour, min, second;
    long nhour, nmin, nsecond;
    time_t now;
    while(!kbhit())
    {
        now = time(NULL);
        second = difftime(now, start);
        hour = (second / 3600.);
        nhour = (long) hour;
        min = fmod(second, 3600.) / 60;
        nmin = (long) min;
        second = fmod(second, 60);
        nsecond = (long) second;
        display_time(nhour, nmin, nsecond);
        sleep(1); // 1초마다 화면을 갱신
    }
}

void display_time(long nhour, long nmin, long nsecond)
{
    system("clear"); // MacOS에서는 "cls" 대신 "clear" 사용
    printf("stopwatch\n\n");
    printf("%.2ld시 %.2ld분 %.2ld초 \n\n", nhour, nmin, nsecond);
    printf("Enter:(시작/정지), SPACE:제시작, Esc:종료\n");
}
```

코드 변경한 부분

`conio.h` 와 그와 관련된 `kbhit()` 와 `getch()` 는 macOS에서 호환이 되지 않아서 제거하고 macOS에서 동작하는 `kbhit()` 함수의 구현을 추가했습니다. 또한 `getch()` 는 표준 C 라이브러리의 `getchar()` 로 변경했습니다.

```
stopwatch
00시 00분 00초
Enter: (시작 / 정지), SPACE: 재시작, Esc: 종료
```

```
stopwatch
00시 00분 06초
Enter: (시작 / 정지), SPACE: 재시작, Esc: 종료
```

출처

- <https://en.m.wikipedia.org/wiki/Conio.h>
- <https://m.blog.naver.com>
- <https://chat.openai.com>