

Ký tự xuống dòng

# Ký tự xuống dòng trên các hệ điều hành

Mac**CR**

Nguyen Van A**CR**

22**CR**

7.4**CR**

Tran Thi B**CR**

21**CR**

9.5**CR**

Windows**LF**

Nguyen Van A**LF**

22**LF**

7.4**LF**

Tran Thi B**LF**

21**LF**

9.5**LF**

Unix**LF**

Nguyen Van A**LF**

22**LF**

7.4**LF**

Tran Thi B**LF**

21**LF**

9.5**LF**

- CR= '\r' và LF='\n'

Để chạy được trên các hệ điều hành khác nhau thì một số phần mềm soạn thảo cho luôn cả 2 ký tự

Nguyen Van A C R L F

22 C R L F

7 . 4 C R L F

Tran Thi B C R L F

21 C R L F

9 . 5 C R L F



Vấn đề là chúng ta không biết lúc testcase dùng `\n` hay `\r` hay `\r\n` làm dấu xuống dòng

# fgets

- Hàm này đọc 1 xâu đọc luôn cả ký tự xuống dòng
- Hàm này có ưu điểm là có thể đọc được dấu hiệu kết thúc đầu vào từ các test-case (input từ file trên HUSTACK, vì chúng ta không thể gõ từ bàn phím) và trả về NULL:  
<https://www.ibm.com/docs/en/i/7.4.0?topic=functions-fgets-read-string>
- Vì thế trong các bài tập trên Hustack, nếu không có dấu hiệu kết thúc thì chúng ta hãy nghĩ đến dùng fgets để đọc đến khi gặp NULL.
- trong khi scanf sẽ đợi cho đến khi chúng ta gõ vào từ bàn phím. Việc này khiến scanf có thể đợi cả đời không thấy chúng ta gõ bàn phím. Như vậy sẽ gặp lỗi TLE (time limit error) trên hustack

# getchar()

- Hàm này cũng có khả năng đọc dấu hiệu kết thúc file và trả về EOF
- `#include <stdio.h>`

```
int main() {
    int c;

    printf("Enter a character: ");
    c = getchar();

    if (c == EOF) {
        printf("End of file or error occurred.\n");
    } else {
        printf("You entered: %c (ASCII value: %d)\n", c, c);
    }
    return 0;
}
```

# scanf khi đọc số

- Khi đọc số mà chúng ta gõ vào từ bàn phím và ấn enter thì scanf để lại một ký tự xuống dòng trong bộ nhớ bàn phím.
- Nếu tiếp theo chúng ta đọc xâu, thì xâu này sẽ chỉ có 1 ký tự xuống dòng

```

int main() {
    Student_t sv1, sv2, sv3;

    // Nhập cho sv1
    fgets(sv1.name, sizeof(sv1.name), stdin);
    sv1.name[strcspn(sv1.name, "\r\n")] = '\0';
    scanf("%d %f", &sv1.age, &sv1.grade); // Đọc cả 2 số cùng lúc cho tiện

    // Xử lý sv2
    sv2 = sv1;
    strcat(sv2.name, " Lorem Ipsum");
    sv2.age += 1;
    sv2.grade -= 1;

    // Nhập cho sv3
    Student_t *pSV = &sv3;

    // Dọn dẹp bộ đệm một cách mạnh mẽ
    int c;
    while ((c = getchar()) != '\n' && c != EOF);

    // Đọc tên cho sv3
    fgets(pSV->name, sizeof(pSV->name), stdin);
    pSV->name[strcspn(pSV->name, "\r\n")] = '\0';

    // Đọc tuổi và điểm cho sv3
    scanf("%d %f", &pSV->age, &pSV->grade);

    // In kết quả
    printf("%s %d %.1f\n", sv1.name, sv1.age, sv1.grade);
    printf("%s %d %.1f\n", sv2.name, sv2.age, sv2.grade);
    printf("%s %d %.1f\n", sv3.name, sv3.age, sv3.grade);

    return 0;
}

```



```
typedef struct {
    char name[MAX_LENGTH];
    int age;
    float grade;
} Student_t;
```

Cách 2 với scanf

```
int main() {
    Student_t sv1, sv2, sv3;
    char c;
    //tren he dieu hanh windows thi dau xuong dong la \n, nhưng tren he dieu hanh ubuntu dau xuong dong co the la \n hoac \r.
    //Nen chung ta phai loai bo ky tu xuong dong nay
    //tren ca tat ca cac he dieu hanh khac nhau. Van de nay chi xay ra khi cac ban doc XAU va ky tu. khong xay ra khi doc so.
    scanf("%[^\\n\\r]%d%f", sv1.name, &sv1.age, &sv1.grade); //Sau khi doc xong sv1.grade, tren mot so chuong trinh dich C de
    //thua lai mot ky tu xuong dong, nen cac ban phai
    //loai bo ky tu xuong dong nay, hoac la \n hoac la \r

    sv2=sv1;
    strcat(sv2.name, " Lorem Ipsum");
    sv2.age +=1;
    sv2.grade -=1;

    while (scanf("%[^\\n\\r]", sv3.name)<=0) //doc bo qua dau xuong dong, chay den khi doc duoc sv3.name thi thoi
        c=getchar(); //doc cai ky tu xuong dong ma scanf("%[^\\n\\r]", sv3.name) khong doc duoc

    scanf("%d%f", &sv3.age, &sv3.grade); //tiếp den la doc sv3.age va sv3.grade

    printf("%s %d %.1f\\n", sv1.name, sv1.age, sv1.grade);
    printf("%s %d %.1f\\n", sv2.name, sv2.age, sv2.grade);
    printf("%s %d %.1f\\n", sv3.name, sv3.age, sv3.grade);

    return 0;
}
```