

3) **character count**

AIM: To develop a c program to generate character count

Procedure : Character-count integrity is a telecommunications term for the ability of a certain link to preserve the number of characters in a message (per unit time, in the case of a user-to-user connection).

Character-count integrity is not the same as character integrity, which requires that the characters delivered be, in fact, exactly the same as they were originated.

Code :

```
#include <stdio.h>
/* count characters and input using while */
main()
{
    long nc;
    nc = 0;
    while (getchar() != EOF)
        ++nc;
    printf("%ld\n", nc); }
```

EXPERIMENT-4

Aim: To implement the data link layer framing method character stuffing.

Problem Description: The character stuffing method gets around the problem of re synchronization after an error by having each frame start and end with special bytes.

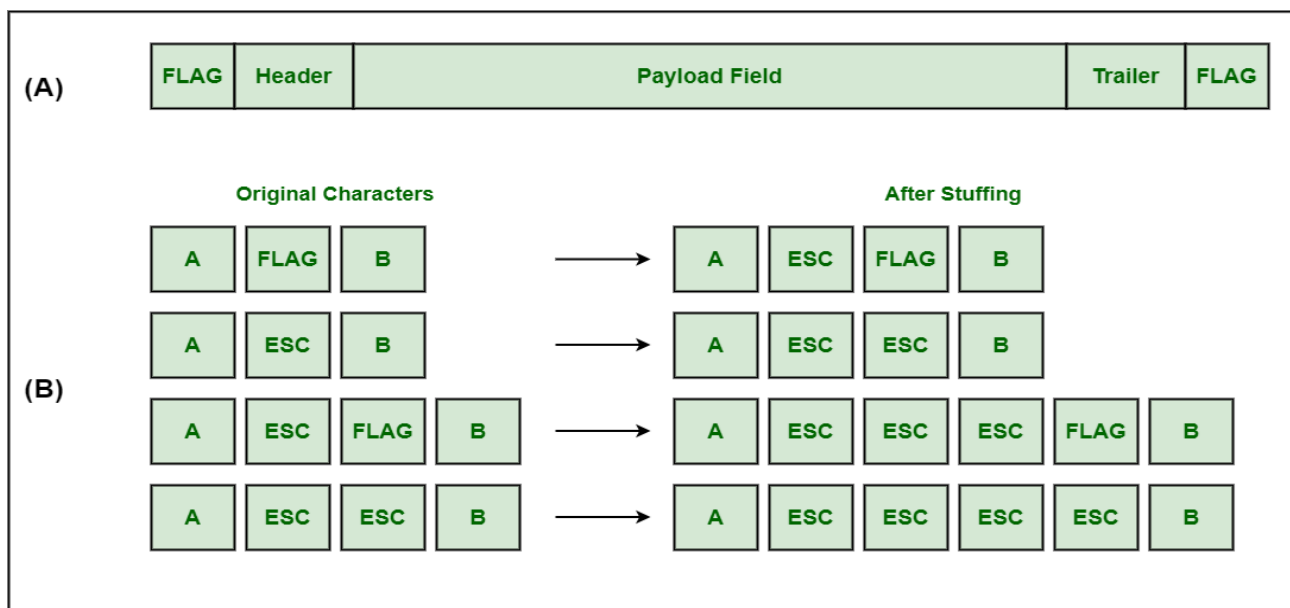
Character Stuffing / Byte Stuffing: Character stuffing or byte stuffing is which an escape byte (ESC) is stuffed character stream before a flag byte in the data.

Explanation: To provide service to network layer, data link layer must use the services provided to it by the physical layer. The bit stream is not guaranteed to be error free. The number of bits received may be less than, equal to, or more than the number of bits transmitted, and they may have different values. It is up to the data link layer to detect and, if necessary, correct errors. The usual approach is for the data link layer to break the bit stream up into discrete frames and compute the checksum for each frame. When a frame arrives at the destination, the checksum is recomputed. If the newly computed checksum is different from one contained in the frame, the data link layer knows that an error has occurred and takes steps to deal with it. In this approach, the “flag byte” is appended at the starting and ending delimiter.

Frame Format:

FLAG	Header	Pay load field	Trailer	FLAG
------	--------	----------------	---------	------

Example: Frame delimited by flag bytes



A Character Stuffing

(A) A frame delimited by flag bytes

(B) Four examples of byte sequences before and after byte stuffing

CharacterStuffing:

```
#include<stdio.h>
int pl=0;
FILE *fp,*fp1;
main()
{
```

```

void stuff();
stuff(); return;
}
void stuff()
{
int H=0,count=0,i=0,c=0,p=0,t=0; char
f[20];
char ch,prev,a[500],se[6]={ 's','t','x','d','l','e'},de[6]={ 'e','t','x','d','l','e'};
printf("enter payload");
scanf("%d",&pl);
printf("enter the file to be stuffed");
scanf("%c",&f);
fp=fopen("c source.txt","r");
fp1=fopen(f1,"w");
L1:while(((fscanf(fp,"%c",&ch1)!=EOF)&&(w!=pl))
{
if(ch=='d')
{
a[H]=ch;
count=count+1;
else
if(ch=='l')
{
a[H]=ch; count=count+1;
}
else if(ch=='e')
{
a[H]=ch; count=count+1;
if(count==3)
{
a[H+1]='d';
a[H+2]='l';
a[H+3]='e';
count=0;
if(H!=pl-3)
{
H=H+3;
count=0;
p=0;
}
else
{
prev=ch;
p=1;
}
}
}
else
{
a[H]=ch;
count=0;
}
}

```

```
H++;
count=0;
if(feof(fp))
c=1;
else
fseek(fp,-1,1)
L2:for(i=0;i<6;i++)
fprintf(fp,"%c",se[i]);
for(i=0;i<H;i++)
fprintf(fp1,"%c",a[i]);
for(i=0;i<6;i++)
fprintf(fp,"%c",de[i]);
fprintf(fp,"\n");
H=0; if(p==1)
{
goto L1;
}
if(c==0)
{
p=0;
goto L1;
} i
f(p==1)
{
p=0;
goto L2;
}
fcloseall();
}
```