Delhi Technological University

Compiler Design Lab : COE-317

Submitted to:
Dr. Rajni Jindal

Submitted by :
Feroz Ahmad
2K14/CO/034
Computer Engineering
B. Tech.

Contents

Experiment 1

Aim

Listing 1: Sample Perl Script With Highlighting

```
/*****************************
FLAG - @
ESC - !
*****************************/
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MSGSZ    1024
#define FLAG '@'
#define ESC '!'

struct message_buf
{
    long    mtype;
    char s[MSGSZ];
};
void charCount(char* s,int a);
void byteStuffing(char* s,int a);

int main()
{
    int msqid;
    int msgflg = 0666;
    key_t key;
    size_t buf_length;
    //buffer struct
    struct message_buf sbuf;

    key = 2929;

    if((msqid = msgget(key, msgflg )) < 0)
    {
        perror("msgget");
        exit(1);
    }
    else
        printf("connected\n");

    char c;
    printf("Press 1 for character count\n Press 2 for Byte Stuffing \n");
    scanf("%c",&c);
    if(msgrcv(msqid, &sbuf, MSGSZ, 1, 0) < 0)
    {
        perror("msgrcv");
        exit(1);
```

Experiment 1 continued on next page...

```
    }

    printf("Message:  Received\n");
    buf_length = strlen(sbuf.s);
    if(c == '1')
    {
        charCount(sbuf.s, buf_length);
    }
    else
    {
        byteStuffing(sbuf.s, buf_length);
    }
    return 0;
}

void charCount(char* s,int len)
{
    int fsize,i,j;
    for(i=0; i < len ; )
    {
        fsize=(s[i]-'0');
        printf("Frame :\t");
        for(j=i+1; j < i + fsize; j++)
        {
            printf("%c",s[j]);
        }
        printf("\n");

        i += fsize;
    }
}
void byteStuffing(char* s,int len)
{
    int fflag = 0, eflag = 0;;
    int i = 0;
    char t;
    for(i = 0; i < len; i++)
    {
        t = s[i];

        if(eflag == 1)
        {
            //ignore
            printf("%c", t);
            eflag = 0;
        }
        else if(eflag == 0 && t == ESC)
        {
            eflag = 1;
        }
        else if(t == FLAG && fflag == 1) //ending flag
        {
            fflag = 0;
```

```
            printf("\n");
        }
        else if(t == FLAG && fflag == 0) //starting flag
        {
            fflag = 1;
            printf("Frame :\t");
        }
        else
        {
            printf("%c", t);
        }
    }
}
```

Screenshot

Experiment 2

Aim

Listing 2: Sample Perl Script With Highlighting

```
/*****************************
FLAG - @
ESC - !
*****************************/
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MSGSZ    1024
#define FLAG '@'
#define ESC '!'

struct message_buf
{
    long    mtype;
    char s[MSGSZ];
};
void charCount(char* s,int a);
void byteStuffing(char* s,int a);

int main()
{
    int msqid;
    int msgflg = 0666;
    key_t key;
    size_t buf_length;
    //buffer struct
    struct message_buf sbuf;

    key = 2929;

    if((msqid = msgget(key, msgflg )) < 0)
    {
        perror("msgget");
        exit(1);
    }
    else
        printf("connected\n");

    char c;
    printf("Press 1 for character count\n Press 2 for Byte Stuffing \n");
    scanf("%c",&c);
    if(msgrcv(msqid, &sbuf, MSGSZ, 1, 0) < 0)
    {
        perror("msgrcv");
        exit(1);
```

Experiment 2 continued on next page...

```
    }

    printf("Message:  Received\n");
    buf_length = strlen(sbuf.s);
    if(c == '1')
    {
        charCount(sbuf.s, buf_length);
    }
    else
    {
        byteStuffing(sbuf.s, buf_length);
    }
    return 0;
}

void charCount(char* s,int len)
{
    int fsize,i,j;
    for(i=0; i < len ; )
    {
        fsize=(s[i]-'0');
        printf("Frame :\t");
        for(j=i+1; j < i + fsize; j++)
        {
            printf("%c",s[j]);
        }
        printf("\n");

        i += fsize;
    }
}
void byteStuffing(char* s,int len)
{
    int fflag = 0, eflag = 0;;
    int i = 0;
    char t;
    for(i = 0; i < len; i++)
    {
        t = s[i];

        if(eflag == 1)
        {
            //ignore
            printf("%c", t);
            eflag = 0;
        }
        else if(eflag == 0 && t == ESC)
        {
            eflag = 1;
        }
        else if(t == FLAG && fflag == 1) //ending flag
        {
            fflag = 0;
```

```
            printf("\n");
        }
        else if(t == FLAG && fflag == 0) //starting flag
        {
            fflag = 1;
            printf("Frame :\t");
        }
        else
        {
            printf("%c", t);
        }
    }
}
```

Screenshot

Experiment 3

Aim

Listing 3: Sample Perl Script With Highlighting

```
/*****************************
FLAG - @
ESC - !
*****************************/
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MSGSZ    1024
#define FLAG '@'
#define ESC '!'

struct message_buf
{
    long    mtype;
    char s[MSGSZ];
};
void charCount(char* s,int a);
void byteStuffing(char* s,int a);

int main()
{
    int msqid;
    int msgflg = 0666;
    key_t key;
    size_t buf_length;
    //buffer struct
    struct message_buf sbuf;

    key = 2929;

    if((msqid = msgget(key, msgflg )) < 0)
    {
        perror("msgget");
        exit(1);
    }
    else
        printf("connected\n");

    char c;
    printf("Press 1 for character count\n Press 2 for Byte Stuffing \n");
    scanf("%c",&c);
    if(msgrcv(msqid, &sbuf, MSGSZ, 1, 0) < 0)
    {
        perror("msgrcv");
        exit(1);
```

```
    }

    printf("Message:  Received\n");
    buf_length = strlen(sbuf.s);
    if(c == '1')
    {
        charCount(sbuf.s, buf_length);
    }
    else
    {
        byteStuffing(sbuf.s, buf_length);
    }
    return 0;
}

void charCount(char* s,int len)
{
    int fsize,i,j;
    for(i=0; i < len ; )
    {
        fsize=(s[i]-'0');
        printf("Frame :\t");
        for(j=i+1; j < i + fsize; j++)
        {
            printf("%c",s[j]);
        }
        printf("\n");

        i += fsize;
    }
}
void byteStuffing(char* s,int len)
{
    int fflag = 0, eflag = 0;;
    int i = 0;
    char t;
    for(i = 0; i < len; i++)
    {
        t = s[i];

        if(eflag == 1)
        {
            //ignore
            printf("%c", t);
            eflag = 0;
        }
        else if(eflag == 0 && t == ESC)
        {
            eflag = 1;
        }
        else if(t == FLAG && fflag == 1) //ending flag
        {
            fflag = 0;
```

```
            printf("\n");
        }
        else if(t == FLAG && fflag == 0) //starting flag
        {
            fflag = 1;
            printf("Frame :\t");
        }
        else
        {
            printf("%c", t);
        }
    }
}
```

Screenshot