

Experiment 5

Objective: Implement various sorting technique used in buffers.

Buffer is a region of memory used to temporarily hold data while it is being moved from one place to another. A buffer is used when moving data between processes within a computer. Majority of buffers are implemented in software. Buffers are generally used when there is a difference between the rate at which data is received and the rate at which it can be processed. Basically the frame are sent from the sender side by assigning a frame id, which could be a number. During the transmission of frames across the link the frames can be transmitted out of order w.r.t the frame id assigned to each of the frame.

The frames need to be in order to maintain integrity. Even though the frames are sent in order, during the transmission the frames may experience delay or routing or any other event which can shuffle the order. Thus frame sorting is done at the receiver side at the buffer at the DATA Link layer

```
#include<stdio.h>
#include<string.h>
#define FRAM_TXT_SIZ 3
#define MAX_NOF_FRAM 127
char str[FRAM_TXT_SIZ*MAX_NOF_FRAM];
struct frame // structure maintained to hold frames
{ char text[FRAM_TXT_SIZ];
int seq_no;
}fr[MAX_NOF_FRAM], shuf_ary[MAX_NOF_FRAM];
int assign_seq_no() //function which splits message
{ int k=0,i,j; //into frames and assigns sequence no
for(i=0; i < strlen(str); k++)
{ fr[k].seq_no = k;
for(j=0; j < FRAM_TXT_SIZ && str[i]!='\0'; j++)
fr[k].text[j] = str[i++];
}
printf("\nAfter assigning sequence numbers:\n");
for(i=0; i < k; i++)
printf("%d:%s ",i,fr[i].text);
return k; //k gives no of frames
}
void generate(int *random_ary, const int limit) //generate array of random nos
{ int r, i=0, j;
while(i < limit)
{ r = random() % limit;
for(j=0; j < i; j++)
if( random_ary[j] == r )
break;
if( i==j ) random_ary[i++] = r;
} }
void shuffle( const int no_frames ) //function shuffles the frames
{
int i, k=0, random_ary[no_frames];
generate(random_ary, no_frames);
for(i=0; i < no_frames; i++)
shuf_ary[i] = fr[random_ary[i]];
printf("\n\nAFTER SHUFFLING:\n");
for(i=0; i < no_frames; i++)
printf("%d:%s ",shuf_ary[i].seq_no,shuf_ary[i].text);
}
void sort(const int no_frames) // sorts the frames
{
```

```

int i,j,flag=1;
struct frame hold;
for(i=0; i < no_frames-1 && flag==1; i++) // search for frames in sequence
{
    flag=0;
    for(j=0; j < no_frames-1-i; j++) // (based on seq no.) and display
    if(shuf Ary[j].seq_no > shuf Ary[j+1].seq_no)
    {
        hold = shuf Ary[j];
        shuf Ary[j] = shuf Ary[j+1];
        shuf Ary[j+1] = hold;
        flag=1;
    }
}
}

int main()
{
int no_frames,i;
printf("Enter the message: ");
gets(str);
no_frames = assign_seq_no();
shuffle(no_frames);
sort(no_frames);
printf("\n\nAFTER SORTING\n");
for(i=0;i<no_frames;i++)
printf("%s",shuf Ary[i].text);
printf("\n\n");
}

```

Enter the message: Welcome to Jodhpur Institute of Technology

After assigning sequence numbers:

0:Wel 1:com 2:e t 3:o J 4:odh 5:pur 6: In 7:sti 8:tut 9:e o 10:f T 11:ech
12:nol 13:ogy

AFTER SHUFFLING:

1:com 4:odh 9:e o 5:pur 3:o J 10:f T 2:e t 6: In 11:ech 13:ogy 0:Wel 8:tut
12:nol 7:sti

AFTER SORTING

Welcome to Jodhpur Institute of Technology