# Inter Process Communication

```
#include<stdlib.h>
#include<stdio.h>
#include<string.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
int main()
{
    int fd[10],child,len;
    char a[20],b[20];
    scanf("%s",a);
    len=strlen(a);
    pipe(fd);
    child=fork();
    if(!child)
    {
        close(fd[0]);
        write(fd[1],a,len);
        wait(0);
    }
    else
    {
        close(fd[1]);
        read(fd[0],b,len);
        printf("string recieved from pipe is : %s",b);
    }
    return 0;
}
```

# System Call

## Integer file style

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
#include<sys/stat.h>
#include<fcntl.h>
int main()
{
   int f1,f2,i=0;
   char c,str[100],b[20];

f1=open("data",O_RDWR|O_CREAT|O_TRUNC
);
   fgets(str,100,stdin);
   i=strlen(str);
   write(f1,str,i);
   close(f1);

   f2=open("data",O_RDONLY);
   read(f2,b,i);
   b[i]='\0';
   printf("\n%s\n",b);
   close(f2);

}
```

**(Shorter program)**

## File pointer(binary file) style

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main()
{
   FILE *f1;
   int i;
   char c,str[100],b[20];
   f1=fopen("data","w+");

   fgets(str,100,stdin);
   i=strlen(str);
   fprintf(f1,"%s",str);

   fseek(f1,0,SEEK_SET);

   fscanf(f1,"%s",b);
   printf("\n%s\n",b);
   fclose(f1);

}
```

# Banker’s Algorithm

```
#include<stdio.h>
int main()
{
   //inputs
   int np,nr,i,j;
   printf("enter no of process and no of
resources:");
   scanf("%d %d",&np,&nr);
   int max[np][nr],alloc[np][nr],need[np][nr];
   int totres[nr],avail[nr],flag[np];
   printf("enter total no of res\n");
   for(i=0;i<nr;i++)
   {
      printf("resource r%d ",i+1);
      scanf("%d",&totres[i]);
   }
   printf("enter avail\n");
   for(i=0;i<nr;i++)
   {
      printf("resource r%d ",i+1);
      scanf("%d",&avail[i]);
   }
   for(i=0;i<np;i++)
   {
      flag[i]=0;
   }
   printf("enter max\n");
   for(i=0;i<np;i++)
   {
      printf("process p%d  ",i);
      for(j=0;j<nr;j++)
      {
         scanf("%d",&max[i][j]);
      }
   }
   printf("enter alloc\n");
   for(i=0;i<np;i++)
   {
      printf("process p%d  ",i);
      for(j=0;j<nr;j++)
      {
         scanf("%d",&alloc[i][j]);
      }
   }
for(i=0;i<np;i++)
{
   for(j=0;j<nr;j++)
   {
      need[i][j]=max[i][j]-alloc[i][j];
   }
}


//working
int safe[np],count=0,pos=0;
while(count!=np)
{
   for(i=0;i<np;i++)
   {
      int f=0;
      if(flag[i]==0)
      {
         for(j=0;j<nr;j++)
         {
            if(need[i][j]<=avail[j]){
            }
            else
            {
               f=1;
            }
         }
         if (f==0)
         {
            for(j=0;j<nr;j++)
            {
               avail[j]=avail[j]+alloc[i][j];
            }
            safe[pos]=i;
            pos++;
            flag[i]=1;
            count++;
         }
      }
   }
}
printf("available ");
for(i=0;i<nr;i++)
{
        printf("%d ",avail[i]);
    }
    printf("\ntotal resource ");
    for(i=0;i<nr;i++)
    {
        printf("%d ",totres[i]);
    }
    printf("\nsafe sequence\n");
    for(i=0;i<np;i++)
    {
        printf("p%d->",safe[i]);
    }

}
```