**SIMPLEST**

#include<stdio.h>

#include<stdlib.h>

struct node{

char data[200];

struct node \*next;

};

struct receiver{

char receive[200];

};

struct sender{

char send[200];

};

int main()

{

struct sender s;

struct receiver r;

int i,k;

struct node \*f= (struct node\*)malloc(sizeof(struct node));

struct node \*sec=(struct node\*) malloc (sizeof(struct node));

struct node \*t=(struct node\*) malloc(sizeof(struct node));

f->next=sec;

sec->next=t;

t->next=NULL;

struct node \*p=f;

for(i=1;i<=3;i++)

{

printf("\nEnter the message=");

scanf("%s",s.send);

k=0;

while(s.send[k]!='\0')

{

p->data[k]=s.send[k];

k++;

}

p=p->next;

}

display(f);

return 0;

}

void display(struct node \*p)

{

if (p==NULL)

{

printf("NO MESSAGE TO SHOW\n");

return;

}

struct node \*temp;

temp=p;

while(temp!=NULL)

{

printf("MESSAGE IS= %s\n",temp->data);

temp=temp->next;

}

}

**HOP TO HOP**

#include<stdio.h>

#include<stdlib.h>

struct node{

int data;

struct node \*next;

struct node \*pre;

};

struct node \*head=NULL;

struct node \*create(struct node \*);

int main()

{

head=create(head);

return 0;

}

struct node \*create (struct node \*head)

{

struct node \*nw,\*temp;

int r,num;

head=NULL;

printf("Enter the router number=");

scanf("%d",&r);

printf("Enter the data=");

scanf("%d",&num);

nw=(struct node\*)malloc(sizeof(struct node));

nw->data=num;

nw->pre=NULL;

nw->next=NULL;

head=nw;

temp=head;

while(r>0)

{

fflush(stdin);

nw=(struct node \*)malloc(sizeof(struct node));

nw->pre=NULL;

nw->next=NULL;

nw->data=temp->data;

while(temp->next!=NULL)

temp=temp->next;

temp->next=nw;

nw->pre=temp;

printf("HOP TO %d ROUTER\n",r);

r--;

}

while(temp->next!=NULL)

{

temp=temp->next;

}

printf("\nReceived Data = %d",temp->data);

return head;

}

**STOP AND WAIT**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct node{

char data1[100];

char data2[100];

struct node \*next;

struct node \*prev;

};

struct sender{

char send\_msg[100];

};

struct receiver{

char receive\_msg[100];

};

int main()

{

int i=0;

struct sender s;

struct receiver r;

struct node \*first=(struct node\*)malloc(sizeof(struct node));

struct node \*second=(struct node\*)malloc(sizeof(struct node));

struct node \*third=(struct node\*)malloc(sizeof(struct node));

first->next=second;

second->next=third;

third->next=NULL;

third->prev=second;

second->prev=first;

first->prev=NULL;

struct node \*p=first;

for(i=1;i<=3;i++)

{

printf("\nEnter the string in sender side :");

gets(s.send\_msg);

int j=0;

while(s.send\_msg[j]!='\0')

{

p->data1[j]=s.send\_msg[j];

j++;

}

p->data1[j]='\0';

printf("Sender :%s",p->data1);

printf("\nreceiver : %s\n",strcpy(r.receive\_msg,p->data1));

printf("\nEnter the feedback to the sender :");

gets(r.receive\_msg);

j=0;

while(r.receive\_msg[j]!='\0')

{

p->data2[j]=r.receive\_msg[j];

j++;

}

p->data2[j]='\0';

printf("\nReceiver : %s",p->data2);

printf("\nSender : %s\n",strcpy(s.send\_msg,p->data2));

p=p->next;

}

return 0;

}

**MESH**

#include<stdio.h>

int main()

{

int op;

printf("\*MESH TOPPOLOGY\*\n");

do

{

printf("\n 1.Send message from A to B");

printf("\n 2.Send message from A to C");

printf("\n 3.Send message from B to C");

printf("\n Enter the option=");

scanf("%d",&op);

switch(op)

{

case 1: printf("YOU ARE IN MACHINE 1 AND SEDING MESSAGE TO MACHINE 2\n");

full1();

break;

case 2: printf("YOU ARE IN MACHINE 1 AND SEDING MESSAGE TO MACHINE 3\n");

full2();

break;

case 3: printf("YOU ARE IN MACHINE 2 AND SEDING MESSAGE TO MACHINE 3\n");

full3();

break;

}

}

while(op!=4);

return 0;

}

void full1()

{

int i;

do

{

printf("\nEnter 1 for machine-A and 0 for machine-B=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void full2()

{

int i;

do

{

printf("\nEnter 1 for machine-A and 0 for machine-C=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void full3()

{

int i;

do

{

printf("\nEnter 1 for machine-B and 0 for machine-C=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void append(int n)

{

FILE \*f;

char mas[1000];

f= fopen("text.txt","a");

printf("Enter the message= \n");

scanf("%s",&mas);

fprintf(f,"\nMachine [%d]: %s",n,mas);

fclose(f);

}

void read()

{

FILE \*f;

char ch;

char name[30];

f= fopen("text.txt","r");

if(f==NULL)

{

printf("File doesn't exist");

}

else

{

printf("The Message is :\n");

while(!feof(f))

{

ch=fgetc(f);

printf("%c",ch);

}

fclose(f);

}

}

**STAR**

#include<stdio.h>

int main()

{

int m ;

printf("\*\*STAR TOPOLOGY\*\*\n");

do

{

printf("Enter the Machine Number=");

scanf("%d",&m);

switch(m)

{

case 1: append(m);

read();

printf("\nMessage send to Machine %d.\n",m);

break;

case 2: append(m);

read();

printf("\nMessage send to Machine %d.\n",m);

break;

case 3: append(m);

read();

printf("\nMessage send to Machine %d.\n",m);

break;

case 4: append(m);

read();

printf("\nMessage send to Machine %d.\n",m);

break;

}

}

while (m!=5);

return 0;

}

void write()

{

FILE \*f;

char mas[1000];

f=fopen("text.txt","w");

printf("Enter the message= \n");

scanf("%s",&mas);

fflush(stdin);

fputs(mas,f);

fclose(f);

}

void append(int n)

{

FILE \*f;

char mas[1000];

f= fopen("text.txt","a");

printf("Enter the message= \n");

scanf("%s",&mas);

fprintf(f,"\nMachine [%d]: %s",n,mas);

fclose(f);

}

void read()

{

FILE \*f;

char ch;

f= fopen("text.txt","r");

if(f==NULL)

{

printf("File doesn't exist");

}

else

{

printf("The Message is :\n");

while(!feof(f))

{

ch=fgetc(f);

printf("%c",ch);

}

fclose(f);

}

}

**RING**

#include<stdio.h>

int main()

{

int op;

printf("\*RING TOPPOLOGY\*\n");

do

{

printf("\n 1.Send message from A to B");

printf("\n 2.Send message from B to C");

printf("\n 3.Send message from C to D");

printf("\n 3.Send message from D to A");

printf("\n Enter the option=");

scanf("%d",&op);

switch(op)

{

case 1: printf("YOU ARE IN MACHINE A AND SEDING MESSAGE TO MACHINE B\n");

full1();

break;

case 2: printf("YOU ARE IN MACHINE B AND SEDING MESSAGE TO MACHINE C\n");

full2();

break;

case 3: printf("YOU ARE IN MACHINE C AND SEDING MESSAGE TO MACHINE D\n");

full3();

break;

case 4: printf("YOU ARE IN MACHINE D AND SEDING MESSAGE TO MACHINE A\n");

full4();

break;

}

}

while(op!=5);

return 0;

}

void full1()

{

int i;

do

{

printf("\nEnter 1 for machine-A and 0 for machine-B=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void full2()

{

int i;

do

{

printf("\nEnter 1 for machine-B and 0 for machine-C=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void full3()

{

int i;

do

{

printf("\nEnter 1 for machine-C and 0 for machine-D=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void full4()

{

int i;

do

{

printf("\nEnter 1 for machine-D and 0 for machine-A=");

scanf("%d",&i);

if(i==1)

{

append(1);

read();

}

else

{

append(0);

read();

}

}

while(i!=2);

return;

}

void append(int n)

{

FILE \*f;

char mas[1000];

f= fopen("text.txt","a");

printf("Enter the message= \n");

scanf("%s",&mas);

fprintf(f,"\nMachine [%d]: %s",n,mas);

fclose(f);

}

void read()

{

FILE \*f;

char ch;

char name[30];

f= fopen("text.txt","r");

if(f==NULL)

{

printf("File doesn't exist");

}

else

{

printf("The Message is :\n");

while(!feof(f))

{

ch=fgetc(f);

printf("%c",ch);

}

fclose(f);

}

}