

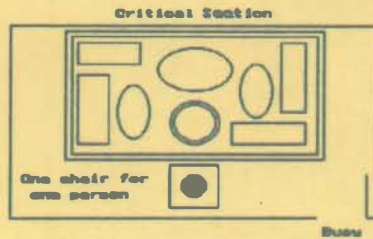
Operating system

Semaphore

Demo

Scan done

$S = 3$
Performing
P operation
if ($a > 0$)
then $a = a - 1$
else (wait on a)



Entrance ->



Exit ->

Md. Khaled Saif Uddin
1997 - 1 - 20 - 007

March 08, 1999

00005
csc

REFERENCE ONLY NOT FOR ISSUE

Operating system

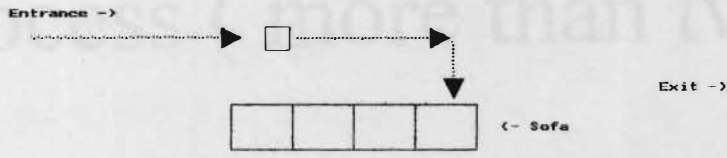
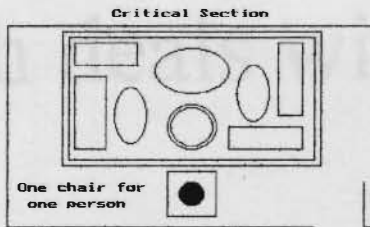
Semaphore



Demo

Report - 07
Scan done.

```
s = 3
Performing
P operation
if ( s > 0 )
then s = s - 1
else (wait on s)
```



Md. Khaled Saif Uddin

1997 - 1 - 20 - 007





Semaphore:

A semaphore is implemented as a protected variable and a process in which processes wait for "V" operations. When a process attempts a "P" operation, it must first decrement the semaphore. If the semaphore is zero, the process must wait until the semaphore is incremented.

P.A. Vennart

operation. The "P" operation on a semaphore S decrements S (S = S - 1). If S is zero, the process must wait until S is incremented. The "V" operation on a semaphore S increments S (S = S + 1).

Example: Consider a semaphore S = 1. Two processes, P1 and P2, are waiting to enter the critical section. P1 is currently in the critical section.

P1 can finish its job in the critical section, change the value of S and do its job in the critical section or wait to finish the critical section.

P2 can enter the critical section if (one or more processes are waiting)

then (let one process proceed)

It means that when a process finishes its job in the critical section then it lets another process (if waiting) to enter critical section, change the value of S and exits.

Accession No. 10000
Call No. 100
Quantity 1

Semaphore :

A semaphore is implemented as a protected variable and a queue in which process can wait for 'V' operations. When a process attempts a 'P' operation on a semaphore whose current value is 0, the process relinquishes the processor and blocks itself to wait a 'V' operation on the semaphore.

P & V operation :

'P' operation : The 'P' operation on the semaphore S, written P(S) operates as follows :

if $S > 0$

then $S = S - 1$

else

(wait on S)

It means that if the value of 'S' is greater than 0 then a process can come in, change the value of 'S' and do his job in the critical section or waits to free the critical section.

'V' operation : The 'V' operation on the semaphore 'S', written V(S) operates as follows :

if (one or more process are waiting)

then (let one of these proceed)

$S = S + 1$

It means that when a process finishes its job in the critical section then it lets another process (if waiting) to enter critical section, change the value of 'S' and exits.

Accession No:- 6869
Call No:
Date: 9.3.79 Price: -

Deaccessed &
Weeded

In this program :-

In this program or demo we have shown an eating room considered as a critical section and a sofa of four seats in front of the eating room which is considered as a semaphore queue. Here we consider the processes as a man. When a man is hungry he comes in and performs the 'P' operation. If he is accepted then he goes in, checks the eating room if it is busy or not. If it is not busy then he goes in and eats his food. If it is busy then he waits on the sofa in front of the eating room. One important thing is when a man is accepted to go in he changes the value of the variable 'S'. When there is no man either in the eating room or on the sofa then the value of s remains full. For example let us consider that there is no one and has the value 5. When a man comes he decreases the value of 'S' and gets in and this process goes on. When the value of 'S' is 0 then no one can enter. When the man in the eating room finishes his job or work he then comes out and performs the 'V' operation. The 'V' operation is if there is any man waiting for the room let one of them proceed. At this time the value of 'S' is incremented. That means there is an empty seat and any man who will come next can get in. If there is no man to come, then the man in the room will let another man proceed after he finishes his job and increments the value of 'S'.

```

#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>

#define mouse 0x33

void start();
void about();
void summary();
void this_program();
void symbol();
void visit();
void mouse_ini();
void page_ini();
void mouse_info();
void hide_mouse();
void page_0();
void page_1();
void page_2();
void page_3();
void page_4();
void page_5();
void page_6();
void gurantee();
void program_by();
void critical();

int n,y,x,np;

main(void)
{
    int m;

    start();
    about();
    summary();
    this_program();
    symbol();
    visit();

    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

    mouse_ini();
    m=0;
    page_0();

    do{

        mouse_info();
        if (x>520 && x<630 && y>380 && y<470 && n==1)
            m=2;
        if (x>440 && x<510 && y>430 && y<470 && n==1 && np==1)
            page_1();
        if (x>440 && x<510 && y>430 && y<470 && n==1 && np==2)
            page_2();
        if (x>440 && x<510 && y>430 && y<470 && n==1 && np==3)

```

```

page_3();
if (x>440 && x<510 && y>430 && y<470 && n==1 && np==4)
page_4();
if (x>440 && x<510 && y>430 && y<470 && n==1 && np==5)
page_5();
if (x>440 && x<510 && y>430 && y<470 && n==1 && np==6)
page_6();
if (x>440 && x<510 && y>430 && y<470 && n==1 && np==7)
m=2;
if (x>360 && x<430 && y>430 && y<470 && n==1 && np==2)
page_0();
if (x>360 && x<430 && y>430 && y<470 && n==1 && np==3)
page_1();
if (x>360 && x<430 && y>430 && y<470 && n==1 && np==4)
page_2();
if (x>360 && x<430 && y>430 && y<470 && n==1 && np==5)
page_3();
if (x>360 && x<430 && y>430 && y<470 && n==1 && np==6)
page_4();
if (x>360 && x<430 && y>430 && y<470 && n==1 && np==7)
page_5();

if (x>360 && x<510 && y>380 && y<420 && n==1)
page_0();

} while (m!=2);

hide_mouse();
cleardevice();
closegraph();

if (np==7){
    gurantee();
    program_by();
}

return 0;
}

void start()
{

int i,j;

int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

settextstyle(7,0,7);

for (i=-400; i<=50; i++){
    setcolor(WHITE);
    outtextxy(i,100,"Semaphore");
    outtextxy(450-i,200,"Demo");
    delay(8);
    setcolor(BLACK);
    outtextxy(i,100,"Semaphore");
    outtextxy(450-i,200,"Demo");
}

setcolor(WHITE);

```

```

outtextxy(50,100,"Semaphore");
outtextxy(400,200,"Demo");

for (i=1; i<=275; i++)
    for (j=1; j<=5; j++)
    {
        putpixel(305-i,190+j,RED);
        putpixel(300+i,190+j,RED);
        delay(2);
    }

delay(2000);
cleardevice();
closegraph();
}

void about()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

    settextstyle(3,0,4);
    setcolor(11);
    outtextxy(140,30,"This program is about");
    settextstyle(10,0,5);
    setcolor(14);
    outtextxy(135,80,"Semaphore");
    settextstyle(3,0,4);
    setcolor(13);
    outtextxy(155,190,"Which deals with the");
    settextstyle(7,0,4);
    setcolor(10);
    outtextxy(94,270,"n-process (more than two)");
    settextstyle(7,0,4);
    setcolor(10);
    outtextxy(125,350,"critical section problem");
    getch();
    cleardevice();
    closegraph();
}

void summary()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

    outtextxy(10,10,"Critical Section : ");
    outtextxy(80,35,"The part of the program where the shared memory is ac");
    outtextxy(80,50,"is called the critical section.");
    outtextxy(10,70,"Semaphore :");
    outtextxy(80,95,"A semaphore can be implemented as a protected variabl");
    outtextxy(80,110,"a queue in which process can wait for 'V' operations");
    outtextxy(80,125,"a process attempts a 'P' operation on the semaphore");
    outtextxy(80,140,"current value is 0. The process relinquishes the pro");
    outtextxy(80,155,"and blocks itself to wait a 'V' operation on the sem");
    outtextxy(10,270,"Busy waiting :");
    outtextxy(80,290,"While a process is in its critical section any other");
    outtextxy(80,305,"that tries to enter must loop continuously which

```



```

outtextxy(80,320,"waiting. It is clearly a problem in a real multi-prog
outtextxy(80,335,"system and the great disadvantage of busy waiting
outtextxy(80,350,"wastes CPU cycles that some other process might be
outtextxy(80,365,"use it productively.");
outtextxy(80,180,"'P' operation on Semaphore | 'V' operation on Sema
outtextxy(80,195,"    if ( s > 0 )           | if (process are wai
outtextxy(80,210,"    then s = s - 1         | then (one may pro
outtextxy(80,225,"    else (wait on S)       | s = s + 1");
outtextxy(170,400,"Press any key to continue ... ..");
getch();
cleardevice();
closegraph();
}

void this_program()
{

int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

outtextxy(20,150,"IN THIS PROGRAM .. ..");
outtextxy(80,185," In this program we have an eating room which is cons
outtextxy(80,200," as a critical section and outside it there is a sofa
outtextxy(80,215," is considered as a queue. Any man (or process) when
outtextxy(80,230," hungry enters and performs the 'P' operation. If ac
outtextxy(80,245," then he goes in. If both the critical section and the
outtextxy(80,260," is empty then he goes in the eaating room and do his
outtextxy(80,275," If the eating room is beign occupied by another man
outtextxy(80,290," he tkes a sit on the sofa. When the man in the eatin
outtextxy(80,305," finishes his work then he comes out of the eating roo
outtextxy(80,320," performs the 'V' operation. If there is any man wait
outtextxy(80,335," then lets one to go in and he exits. ");
outtextxy(170,400,"press any key to continue ... ..");
getch();
cleardevice();
closegraph();
}

void symbol()
{

int gdriver = DETECT, gmode, errorcode;
initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

settextstyle(3,0,4);
outtextxy(20,30,"Some symbols");
line(20,70,230,70);
rectangle(20,120,40,140);
outtextxy(60,110,"<- denotes a man (or process)");
rectangle(20,180,60,220);
outtextxy(80,180,"<- denotes an empty sit");
rectangle(20,240,60,280);
sector(40,260,0,360,10,10);
outtextxy(80,240,"<- denotes an occupied sit");
outtextxy(20,300,"'S' is a variable who counts how many");
outtextxy(20,350,"man enters and exits");
getch();
cleardevice();
closegraph();
}

```

```

void visit()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

    settextstyle(3,0,4);
    outtextxy(80,30,"We have described semaphore");
    outtextxy(90,110,"and also the symbols used in");
    outtextxy(130,180,"this program. So before");
    outtextxy(100,240,"starting the demo why don't");
    outtextxy(100,300,"we visit the place of critical");
    outtextxy(200,360,"section first ?");
    getch();
    cleardevice();
    closegraph();
}

void mouse_ini()
{
    union REGS regs;

    regs.x.ax = 0;
    int86(mouse, &regs, &regs);
    regs.x.ax = 1;
    int86(mouse, &regs, &regs);
}

void mouse_info()
{
    union REGS regs;

    regs.x.ax = 3;
    int86(mouse, &regs, &regs);
    n=regs.x.bx;
    y=regs.x.dx;
    x=regs.x.cx;
}

void hide_mouse()
{
    union REGS regs;

    regs.x.ax = 2;
    int86(mouse, &regs, &regs);
}

void page_ini()
{
    rectangle(20,380,350,470);
    rectangle(360,380,510,420);
    rectangle(360,430,430,470);
    rectangle(440,430,510,470);
    rectangle(520,380,630,470);
    outtextxy(390,400,"Start Again");
    outtextxy(370,450,"< Back");
    outtextxy(455,450,"Next >");
    outtextxy(555,425,"Exit");
    outtextxy(25,402,"(1) Start Again:Start from the begining.");
}

```

```

    outtextxy(25,417,"(2) Back           :Previous page.");
    outtextxy(25,433,"(3) Next          :Next page.");
    outtextxy(25,448,"(4) Exit         :Exit the program.");
}

void page_0()
{
    int i,j;

    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
    critical();
    np=1;
    n=0;
    mouse_ini();
}

void page_1()
{
    int i,j;
    int s=5;

    n=0;
    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
    critical();
    gotoxy(2,2);
    printf("S = %d",s);
    delay(2000);
    rectangle(10,250,30,271);
    outtextxy(10,40,"Performing");
    outtextxy(10,50,"P operation");
    outtextxy(10,70,"if ( s > 0 )");
    outtextxy(10,80," then s = s - 1");
    outtextxy(10,90,"else (wait on s)");
    delay(2000);
    s=s-1;
    gotoxy(2,2);
    printf("S = %d",s);

    for (i=0; i<=400; i++){          /* one process coming */
        setcolor(WHITE);
        rectangle(10+i,250,10+20+i,251+20);
        delay(20);
        setcolor(BLACK);
        rectangle(10+i,250,10+20+i,251+20);
    }
    setcolor(WHITE);
    rectangle(10+i,250,10+20+i,251+20);

    for (i=251;i>160;i--){          /* going into critical section */
        setcolor(WHITE);
        rectangle(410,i,410+21,i+21);
        delay(20);
        setcolor(BLACK);
        rectangle(410,i,410+21,i+21);
    }
}

```

```

}
setcolor(WHITE);
rectangle(410,i,410+21,i+21);

outtextxy(405,210,"Busy"); /* raising busy flag */

setcolor(BLACK);
outtextxy(10,40,"Performing");
outtextxy(10,50,"P operation");
outtextxy(10,70,"if ( s > 0 )");
outtextxy(10,80," then s = s - 1");
outtextxy(10,90,"else (wait on s)");
setcolor(WHITE);

for (i=410;i>325;i--){ /* going to sit on the chair */
    setcolor(WHITE);
    rectangle(i,159,i+21,181);
    delay(25);
    setcolor(BLACK);
    rectangle(i,159,i+21,181);
}
setcolor(WHITE);
sector(300,170,0,360,10,10); /* on the chair */

delay(2000);

rectangle(10,250,30,271);
outtextxy(10,40,"Performing");
outtextxy(10,50,"P operation");
outtextxy(10,70,"if ( s > 0 )");
outtextxy(10,80," then s = s - 1");
outtextxy(10,90,"else (wait on s)");
delay(2000);
s=s-1;
gotoxy(2,2);
printf("S = %d",s);

for (i=0; i<=355; i++){ /* one process coming */
    setcolor(WHITE);
    rectangle(10+i,250,10+20+i,251+20);
    delay(20);
    setcolor(BLACK);
    rectangle(10+i,250,10+20+i,251+20);
}
setcolor(WHITE);
rectangle(10+i,250,10+20+i,251+20);

for (i=250;i<=290;i++){ /* going to sit */
    setcolor(WHITE);
    rectangle(365,i,365+21,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(365,i,365+21,i+20);
}
setcolor(WHITE);
sector(375,340,0,360,10,10); /* on the 4 sofa */

setcolor(BLACK);
outtextxy(10,40,"Performing");
outtextxy(10,50,"P operation");

```

```

    outtextxy(10,70,"if ( s > 0 )");
    outtextxy(10,80," then s = s - 1");
    outtextxy(10,90,"else (wait on s)");
    setcolor(WHITE);
    np=2;
    n=0;
    mouse_ini();
}

void page_2()
{
    int i,j;
    int s=0;

    n=0;
    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
    critical();
    sector(225,340,0,360,10,10);    /* on the 1 sofa */
    sector(275,340,0,360,10,10);    /* on the 2 sofa */
    sector(325,340,0,360,10,10);    /* on the 3 sofa */
    sector(375,340,0,360,10,10);    /* on the 4 sofa */
    sector(300,170,0,360,10,10);    /* on the chair */
    gotoxy(2,2);
    printf("S = %d",s);
    np=3;
    n=0;
    mouse_ini();
}

void page_3()
{
    int i,j;
    int s=0;

    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
    outtextxy(405,210,"Busy");
    critical();
    sector(225,340,0,360,10,10);    /* on the 1 sofa */
    sector(275,340,0,360,10,10);    /* on the 2 sofa */
    sector(325,340,0,360,10,10);    /* on the 3 sofa */
    sector(375,340,0,360,10,10);    /* on the 4 sofa */
    sector(300,170,0,360,10,10);    /* on the chair */

    gotoxy(2,2);
    printf("S = %d",s);

    outtextxy(470,30,"Process Execution");

    for (i=0; i<=100; i++){
        gotoxy(60,4);
        printf("%d% Complete",i);
        delay(20);
    }
}

```

```

setcolor(BLACK);          /* releazing from the chair */

for (j=10; j>=0; j--)
    for (i=0; i<=360; i++)
        ellipse(300,170,i,0,j,j);

setcolor(WHITE);
rectangle(340,160,360,180);

for (i=340;i<410;i++){
    setcolor(WHITE);
    rectangle(i,160,i+20,180);
    delay(20);
    setcolor(BLACK);
    rectangle(i,160,i+20,180);
}
setcolor(WHITE);
rectangle(i,160,i+20,180);

setcolor(BLACK);          /* removing busy flag */
outtextxy(405,210,"Busy");
setcolor(WHITE);

for (i=160;i<=250;i++){
    setcolor(WHITE);
    rectangle(410,i,410+20,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(410,i,410+20,i+20);
}
setcolor(WHITE);
rectangle(410,i,410+20,i+20);

outtextxy(470,70,"Performing");      /* V operation */
outtextxy(470,80,"V operation");
outtextxy(470,100,"if (waiting)");
outtextxy(470,110," then (one proceed)");
outtextxy(470,120," s = s + 1");
s=s+1;
delay(2000);
gotoxy(2,2);
printf("S = %d",s);

setcolor(BLACK);          /* releazing from the sofa */

for (j=10; j>=0; j--)
    for (i=0; i<=360; i++)
        ellipse(375,340,i,0,j,j);

setcolor(WHITE);
rectangle(365,290,385,310);

for (i=290;i>=252;i--){          /* calling one up */
    setcolor(WHITE);
    rectangle(365,i,365+20,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(365,i,365+20,i+20);
}
setcolor(WHITE);

```

```

rectangle(365,i,365+20,i+20);

for (i=0; i<=45; i++){
    setcolor(WHITE);
    rectangle(410+i,250,410+20+i,251+20);
    rectangle(365+i,251,365+20+i,252+20);
    delay(20);
    setcolor(BLACK);
    rectangle(410+i,250,410+20+i,251+20);
    rectangle(365+i,251,365+20+i,252+20);
}
setcolor(WHITE);
rectangle(410+i,250,410+20+i,251+20);
rectangle(365+i,251,365+20+i,252+20);

for (i=251;i>160;i--){ /* going into critical section */
    setcolor(WHITE);
    rectangle(410,i,410+21,i+21);
    delay(20);
    setcolor(BLACK);
    rectangle(410,i,410+21,i+21);
}
setcolor(WHITE);
rectangle(410,i,410+21,i+21);

outtextxy(405,210,"Busy"); /* raising busy flag */

setcolor(BLACK);
outtextxy(470,70,"Performing"); /* V operation */
outtextxy(470,80,"V operation");
outtextxy(470,100,"if (waiting)");
outtextxy(470,110," then (one proceed)");
outtextxy(470,120," s = s + 1");
setcolor(WHITE);

for (i=455;i<=600;i++){ /* one process exits */
    setcolor(WHITE);
    rectangle(i,250,i+20,271);
    delay(20);
    setcolor(BLACK);
    rectangle(i,250,i+20,271);
}

gotoxy(60,4);
printf("0% Complete ",i);

for (i=410;i>325;i--){ /* going to sit on the chair */
    setcolor(WHITE);
    rectangle(i,159,i+21,181);
    delay(25);
    setcolor(BLACK);
    rectangle(i,159,i+21,181);
}
setcolor(WHITE);
sector(300,170,0,360,10,10); /* on the chair */

np=4;
n=0;
mouse_ini();
}

```

```

void page_4()
{
    int i,j,s;

    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
    outtextxy(405,210,"Busy");
    critical();
    sector(225,340,0,360,10,10);    /* on the 1 sofa    */
    sector(275,340,0,360,10,10);    /* on the 2 sofa    */
    sector(325,340,0,360,10,10);    /* on the 3 sofa    */
    sector(300,170,0,360,10,10);    /* on the chair     */
    s=1;
    gotoxy(2,2);
    printf("S = %d",s);
    delay(2000);
    rectangle(10,250,30,271);
    outtextxy(10,40,"Performing");
    outtextxy(10,50,"P operation");
    outtextxy(10,70,"if ( s > 0 )");
    outtextxy(10,80," then s = s - 1");
    outtextxy(10,90,"else (wait on s)");
    delay(1000);
    s=s-1;
    gotoxy(2,2);
    printf("S = %d",s);

    for (i=0; i<=355; i++){          /* one process comming */
        setcolor(WHITE);
        rectangle(10+i,250,10+20+i,251+20);
        delay(20);
        setcolor(BLACK);
        rectangle(10+i,250,10+20+i,251+20);
    }
    setcolor(WHITE);
    rectangle(10+i,250,10+20+i,251+20);

    for (i=250;i<=290;i++){          /* going to sit      */
        setcolor(WHITE);
        rectangle(365,i,365+21,i+20);
        delay(20);
        setcolor(BLACK);
        rectangle(365,i,365+21,i+20);
    }
    setcolor(WHITE);
    sector(375,340,0,360,10,10);    /* on the 4 sofa     */

    setcolor(BLACK);
    outtextxy(10,40,"Performing");
    outtextxy(10,50,"P operation");
    outtextxy(10,70,"if ( s > 0 )");
    outtextxy(10,80," then s = s - 1");
    outtextxy(10,90,"else (wait on s)");
    setcolor(WHITE);

    delay(2000);
}

```



```

rectangle(10,250,30,271);
outtextxy(10,40,"Performing");
outtextxy(10,50,"P operation");
outtextxy(10,70,"if ( s > 0 )");
outtextxy(10,80," then s = s - 1");
outtextxy(10,90,"else (wait on s)");

delay(1000);

for (i=0; i<=55; i++){          /* one process not entering */
    setcolor(WHITE);
    rectangle(10+i,250,10+20+i,251+20);
    delay(20);
    setcolor(BLACK);
    rectangle(10+i,250,10+20+i,251+20);
}
setcolor(WHITE);
rectangle(10+i,250,10+20+i,251+20);

for (i=250;i<=320;i++){
    setcolor(WHITE);
    rectangle(65,i,65+21,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(65,i,65+21,i+20);
}
setcolor(WHITE);
rectangle(65,i,65+21,i+20);

for (i=55; i>=10; i--){
    setcolor(WHITE);
    rectangle(10+i,320,10+21+i,320+21);
    delay(20);
    setcolor(BLACK);
    rectangle(10+i,320,10+21+i,320+21);
}
setcolor(WHITE);

setcolor(BLACK);
outtextxy(10,40,"Performing");
outtextxy(10,50,"P operation");
outtextxy(10,70,"if ( s > 0 )");
outtextxy(10,80," then s = s - 1");
outtextxy(10,90,"else (wait on s)");
setcolor(WHITE);
np=5;
n=0;
mouse_ini();
}

```

```

void page_5()

```

```

{
    int i,j;
    int s=0;

    n=0;
    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
}

```

```

outtextxy(405,210,"Busy");
critical();
sector(225,340,0,360,10,10); /* on the 1 sofa */
sector(275,340,0,360,10,10); /* on the 2 sofa */
sector(325,340,0,360,10,10); /* on the 3 sofa */
sector(375,340,0,360,10,10); /* on the 4 sofa */
sector(300,170,0,360,10,10); /* on the chair */

gotoxy(2,2);
printf("S = %d",s);

outtextxy(470,30,"Process Execution");

for (i=0; i<=100; i++){
    gotoxy(60,4);
    printf("%d% Complete",i);
    delay(20);
}
setcolor(BLACK); /* releazing from the chair */

for (j=10; j>=0; j--){
    for (i=0; i<=360; i++){
        ellipse(300,170,i,0,j,j);
    }
}

setcolor(WHITE);
rectangle(340,160,360,180);

for (i=340;i<410;i++){
    setcolor(WHITE);
    rectangle(i,160,i+20,180);
    delay(20);
    setcolor(BLACK);
    rectangle(i,160,i+20,180);
}
setcolor(WHITE);
rectangle(i,160,i+20,180);

setcolor(BLACK); /* removing busy flag */
outtextxy(405,210,"Busy");
setcolor(WHITE);

for (i=160;i<=250;i++){
    setcolor(WHITE);
    rectangle(410,i,410+20,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(410,i,410+20,i+20);
}
setcolor(WHITE);
rectangle(410,i,410+20,i+20);

outtextxy(470,70,"Performing"); /* V operation */
outtextxy(470,80,"V operation");
outtextxy(470,100,"if (waiting)");
outtextxy(470,110," then (one proceed)");
outtextxy(470,120," s = s + 1");
s=s+1;
delay(2000);
gotoxy(2,2);
printf("S = %d",s);

```

```

setcolor(BLACK);          /* releasing from the sofa */

for (j=10; j>=0; j--)
    for (i=0; i<=360; i++)
        ellipse(375,340,i,0,j,j);

setcolor(WHITE);
rectangle(365,290,385,310);

for (i=290;i>=252;i--){    /* calling one up */
    setcolor(WHITE);
    rectangle(365,i,365+20,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(365,i,365+20,i+20);
}
setcolor(WHITE);
rectangle(365,i,365+20,i+20);

for (i=0; i<=45; i++){
    setcolor(WHITE);
    rectangle(410+i,250,410+20+i,251+20);
    rectangle(365+i,251,365+20+i,252+20);
    delay(20);
    setcolor(BLACK);
    rectangle(410+i,250,410+20+i,251+20);
    rectangle(365+i,251,365+20+i,252+20);
}
setcolor(WHITE);
rectangle(410+i,250,410+20+i,251+20);
rectangle(365+i,251,365+20+i,252+20);

for (i=251;i>160;i--){    /* going into critical section */
    setcolor(WHITE);
    rectangle(410,i,410+21,i+21);
    delay(20);
    setcolor(BLACK);
    rectangle(410,i,410+21,i+21);
}
setcolor(WHITE);
rectangle(410,i,410+21,i+21);

outtextxy(405,210,"Busy"); /* raising busy flag */

setcolor(BLACK);
outtextxy(470,70,"Performing"); /* V operation */
outtextxy(470,80,"V operation");
outtextxy(470,100,"if (waiting)");
outtextxy(470,110," then (one proceed)");
outtextxy(470,120," s = s + 1");
setcolor(WHITE);

for (i=455;i<=600;i++){    /* one process exits */
    setcolor(WHITE);
    rectangle(i,250,i+20,271);
    delay(20);
    setcolor(BLACK);
    rectangle(i,250,i+20,271);
}

```

```

}

gotoxy(60,4);
printf("0% Complete ",i);

for (i=410;i>325;i--){          /* going to sit on the chair */
    setcolor(WHITE);
    rectangle(i,159,i+21,181);
    delay(25);
    setcolor(BLACK);
    rectangle(i,159,i+21,181);
}
setcolor(WHITE);

sector(300,170,0,360,10,10);    /* on the chair */
np=6;
n=0;
mouse_ini();
}

void page_6()
{
    int i,j;
    int s=1;

    n=0;
    hide_mouse();
    cleardevice();
    delay(1000);
    page_ini();
    outtextxy(405,210,"Busy");
    critical();
    sector(225,340,0,360,10,10); /* on the 1 sofa */
    sector(275,340,0,360,10,10); /* on the 2 sofa */
    sector(325,340,0,360,10,10); /* on the 3 sofa */
    sector(300,170,0,360,10,10); /* on the chair */

    gotoxy(2,2);
    printf("S = %d",s);

    outtextxy(470,30,"Process Execution");

    for (i=0; i<=100; i++){
        gotoxy(60,4);
        printf("%d% Complete",i);
        delay(20);
    }

    setcolor(BLACK);          /* releasing from the chair */

    for (j=10; j>=0; j--)
        for (i=0; i<=360; i++)
            ellipse(300,170,i,0,j,j);

    setcolor(WHITE);
    rectangle(340,160,360,180);

    for (i=340;i<410;i++){
        setcolor(WHITE);

```

```

    rectangle(i,160,i+20,180);
    delay(20);
    setcolor(BLACK);
    rectangle(i,160,i+20,180);
}
setcolor(WHITE);
rectangle(i,160,i+20,180);

setcolor(BLACK);          /* removing busy flag */
outtextxy(405,210,"Busy");
setcolor(WHITE);

for (i=160;i<=250;i++){
    setcolor(WHITE);
    rectangle(410,i,410+20,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(410,i,410+20,i+20);
}
setcolor(WHITE);
rectangle(410,i,410+20,i+20);

outtextxy(470,70,"Performing");          /* V operation */
outtextxy(470,80,"V operation");
outtextxy(470,100,"if (waiting)");
outtextxy(470,110," then (one proceed)");
outtextxy(470,120," s = s + 1");
s=s+1;
delay(2000);
gotoxy(2,2);
printf("S = %d",s);

setcolor(BLACK);          /* releazing from the sofa */

for (j=10; j>=0; j--){
    for (i=0; i<=360; i++){
        ellipse(325,340,i,0,j,j);
    }
}

setcolor(WHITE);
rectangle(315,290,335,310);

for (i=290;i>=252;i--){          /* calling one up */
    setcolor(WHITE);
    rectangle(315,i,315+20,i+20);
    delay(20);
    setcolor(BLACK);
    rectangle(315,i,315+20,i+20);
}
setcolor(WHITE);
rectangle(315,i,315+20,i+20);

for (i=0; i<=45; i++){
    setcolor(WHITE);
    rectangle(410+i,250,410+20+i,251+20);
    rectangle(315+i,251,315+20+i,252+20);
    delay(20);
    setcolor(BLACK);
    rectangle(410+i,250,410+20+i,251+20);
    rectangle(315+i,251,315+20+i,252+20);
}

```

```

setcolor(WHITE);
rectangle(410+i,250,410+20+i,251+20);
rectangle(315+i,251,315+20+i,252+20);

for (i=45; i<=95; i++){
    setcolor(WHITE);
    rectangle(315+i,251,315+20+i,252+20);
    delay(20);
    setcolor(BLACK);
    rectangle(315+i,251,315+20+i,252+20);
}
setcolor(WHITE);
rectangle(315+i,251,315+20+i,252+20);

for (i=251;i>160;i--){ /* going into critical section */
    setcolor(WHITE);
    rectangle(410,i,410+21,i+21);
    delay(20);
    setcolor(BLACK);
    rectangle(410,i,410+21,i+21);
}
setcolor(WHITE);
rectangle(410,i,410+21,i+21);

outtextxy(405,210,"Busy"); /* raising busy flag */

setcolor(BLACK);
outtextxy(470,70,"Performing"); /* V operation */
outtextxy(470,80,"V operation");
outtextxy(470,100,"if (waiting)");
outtextxy(470,110," then (one proceed)");
outtextxy(470,120," s = s + 1");
setcolor(WHITE);

for (i=455;i<=600;i++){ /* one process exits */
    setcolor(WHITE);
    rectangle(i,250,i+20,271);
    delay(20);
    setcolor(BLACK);
    rectangle(i,250,i+20,271);
}

gotoxy(60,4);
printf("0% Complete ",i);

for (i=410;i>325;i--){ /* going to sit on the chair */
    setcolor(WHITE);
    rectangle(i,159,i+21,181);
    delay(25);
    setcolor(BLACK);
    rectangle(i,159,i+21,181);
}
setcolor(WHITE);
sector(300,170,0,360,10,10); /* on the chair */

np=7;
n=0;
mouse_ini();

```

```

}

void gurantee()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

    settextstyle(3,0,4);
    outtextxy(80,30,"So far we have shown that how");
    outtextxy(80,90,"semaphore gurantee's more than");
    outtextxy(80,150,"two processess to access the");
    outtextxy(80,210,"critical section. It also overcomes");
    outtextxy(80,270,"the need for busy waiting.");
    getch();
    cleardevice();
    closegraph();
}

```

```

void program_by()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, "c:\\tcc\\bgi");

    settextstyle(3,0,4);
    setcolor(11);
    outtextxy(180,30," Programmed by");
    settextstyle(10,0,4);
    setcolor(14);
    outtextxy(60,80,"Md. Khaled Saif Uddin");
    settextstyle(3,0,5);
    setcolor(13);
    outtextxy(70,190,"1997 - 1 - 20 - 007");
    settextstyle(7,0,5);
    setcolor(14);
    outtextxy(94,270,"East West University");
    settextstyle(7,0,5);
    setcolor(14);
    outtextxy(40,350,"Dept. of Computer Science");
    getch();
    cleardevice();
    closegraph();
}

```

```

void critical()
{
    outtextxy(236,5,"Critical Section");
    rectangle(150,20,450,200);
    line(440,160,440,200);
    setcolor(BLACK);
    line(400,200,440,200);
    setcolor(WHITE);
    rectangle(200,30,400,140);      /* A Table */
    rectangle(195,25,405,145);      /* A Table */
    circle(300,110,20);
    circle(300,110,17);
    ellipse(300,60,360,0,30,20);
    ellipse(250,100,360,0,13,25);
    ellipse(350,80,360,0,13,25);
    rectangle(205,35,255,55);
    rectangle(205,65,230,130);
}

```

```
rectangle(330,110,390,130);
rectangle(370,35,390,100);
rectangle(280,150,320,190);    /* A Chair */
rectangle(200,320,400,360);    /* Sofa */
line(250,320,250,360);
line(300,320,300,360);
line(350,320,350,360);
outtextxy(20,235,"Entrance ->");
outtextxy(550,300,"Exit ->");
outtextxy(420,336,"<- Sofa");
outtextxy(160,160,"One chair for");
outtextxy(160,175," one person");
}
```

