

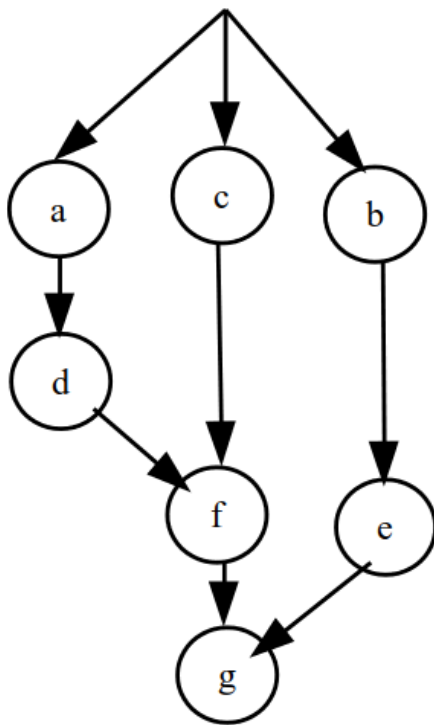
Febrero 2016

Indique el grafo de precedencias que correspondería al siguiente programa.

<i>Program P</i>	<i>process P1</i>	<i>process P2</i>	<i>process P3</i>	<i>begin</i>
<i>var</i>	<i>begin</i>	<i>begin</i>	<i>begin</i>	<i>initial(s1,2);</i>
<i>s1: semaphore;</i>	<i>a;</i>	<i>wait(s1);</i>	<i>wait(s1);</i>	<i>initial(s2,1);</i>
<i>s2: semaphore;</i>	<i>signal(s1);</i>	<i>c;</i>	<i>b;</i>	<i>initial(s3,0);</i>
<i>s3: semaphore;</i>	<i>signal(s1);</i>	<i>signal(s2);</i>	<i>signal(s2);</i>	<i>initial(s4,0);</i>
<i>s4: semaphore;</i>	<i>wait(s2);</i>	<i>wait(s3);</i>	<i>e;</i>	<i>conbegin</i>
	<i>d;</i>	<i>f;</i>	<i>signal(s4);</i>	<i>P1; P2; P3;</i>
	<i>signal(s3);</i>	<i>wait(s4);</i>	<i>end</i>	<i>coend</i>
	<i>signal(s3);</i>	<i>g;</i>		<i>end</i>
	<i>end</i>	<i>end</i>		

Febrero 2016

SOLUCIÓN



Febrero 2017

Indique el grafo de precedencias que correspondería al siguiente programa.

Program P

var

s1: semaphore;

s2: semaphore;

s3: semaphore;

s4: semaphore;

s4: semaphore;

process P1

begin

A;

signal(s1);

wait(s5);

wait(s5);

G;

end

process P2

begin

wait(s1);

B;

signal(s2);

C

signal(s3)

wait(s4);

F;

signal(s5);

end

process P3

begin

wait(s2);

D;

signal(s4);

wait(s3);

E;

signal(s5);

end

begin

initial(s1,0);

initial(s2,0);

initial(s3,0);

initial(s4,0);

initial(s5,2);

conbegin

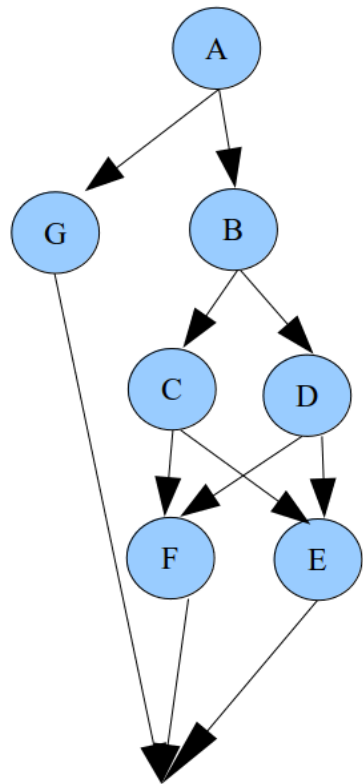
P1; P2; P3;

coend

end

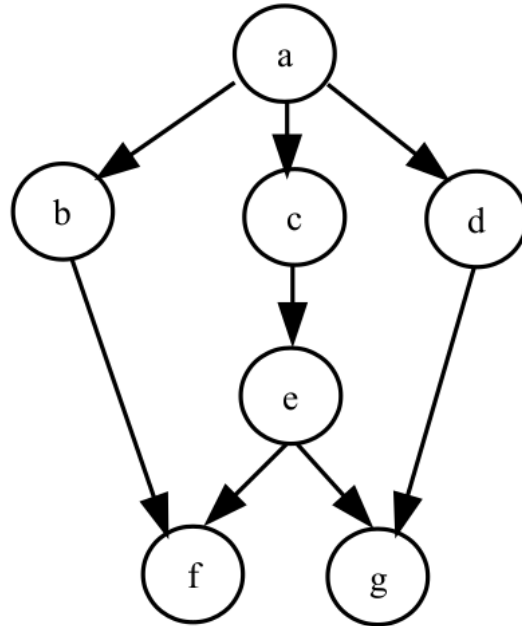
Febrero 2017

SOLUCIÓN



Septiembre 2017

Dado el siguiente programa, corríjalo, usando semáforos, para que se cumpla el grafo de precedencias que se indica. No olvide inicializar los semáforos que use



```
Program P
var

process P1
begin
    a;
    d;
end
process P2
begin
    c;
    e;
    g;
end
process P3
begin
    b;
    f;
end
begin
    conbegin
        P1; P2; P3;
    coend
end
```

Septiembre 2017

SOLUCIÓN

Program P

```
var  
    sa, sb, sd, se: semaphore;
```

process P1

```
begin  
    a;  
    signal(sB);  
    signal(sC);  
    d;  
    signal(sG);  
end
```

process P2

```
begin  
    wait(sC);  
    c;  
    e;  
    signal(sF);  
    wait(sG);  
    g;  
end
```

process P3

```
begin  
    wait(sB);  
    b;  
    WAIT(sF);  
    f;  
end
```

begin

```
    initial(sB, 0);  
    initial(sC, 0);  
    initial(sG, 0);  
    initial(sF, 0);  
    conbegin  
        P1; P2; P3;
```

end

```
    coend
```

Septiembre 2017

SOLUCIÓN

Program P

```
var  
    sa,sb,sd,se:semaphore;
```

process P1

```
begin  
    a;  
    signal(sB);  
    signal(sC);  
    d;  
    signal(sG);  
end
```

process P2

```
begin  
    wait(sC);  
    c;  
    e;  
    signal(sF);  
    wait(sG);  
    g;  
end
```

process P3

```
begin  
    wait(sB);  
    b;  
    WAIT(sF);  
    f;  
end
```

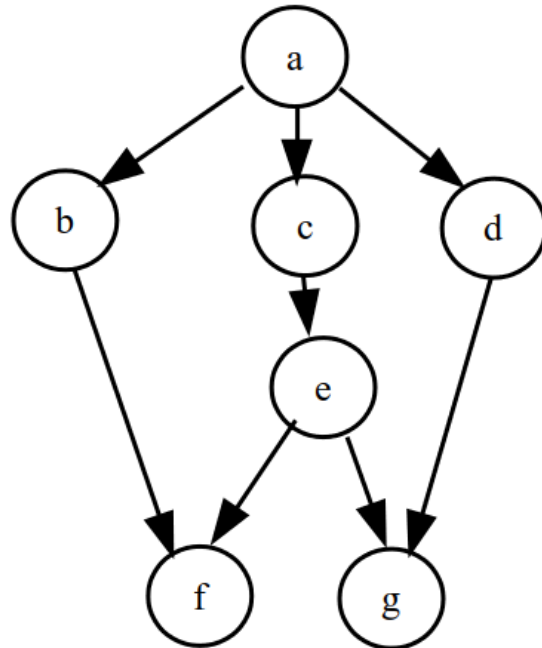
begin

```
    initial(sB,0);  
    initial(sC,0);  
    initial(sG,0);  
    initial(sF,0);  
    conbegin  
        P1; P2; P3;
```

end

Septiembre 2018

Dado el siguiente programa, corríjalo, usando semáforos, para que se cumpla el grafo de precedencias que se indica. No olvide inicializar los semáforos que use



```
Program P  
var  
  
process P1  
begin  
    a;  
    b;  
    f;  
end  
  
process P2  
begin  
    c;  
    e;  
end  
  
process P3  
begin  
    d;  
    g;  
end  
  
begin  
    conbegin  
        P1; P2; P3;  
    coend  
end
```


Septiembre 2018

SOLUCIÓN

Program P

```
var  
sa, sb, sd, se: semaphore
```

process P1

```
begin  
    a;  
    signal(s1);  
    signal(s1);  
    b;  
    wait(s2);  
    f;  
end
```

process P2

```
begin  
    wait(s1);  
    c;  
    e;  
    signal(s2);  
    signal(s3);  
end
```

process P3

```
begin  
    wait(s1);  
    d;  
    wait(s3);  
    g;  
end
```

begin

```
    initial(s1,0);  
    initial(s2,0);  
    initial(s3,0);  
  
    conbegin  
        P1; P2; P3;  
    coend  
end
```

Septiembre 2019

Indique el grafo de precedencias que correspondería al siguiente programa

Program P

```
var  
  sc:semaphore;  
  sd:semaphore;  
  se:semaphore;
```

process P1

```
begin  
  a;  
  signal(sc);  
  signal(sd);  
  b;  
  wait(se);  
  e;  
  signal(sc);  
end
```

process P2

```
begin  
  wait(sd);  
  d;  
  f;  
end
```

process P3

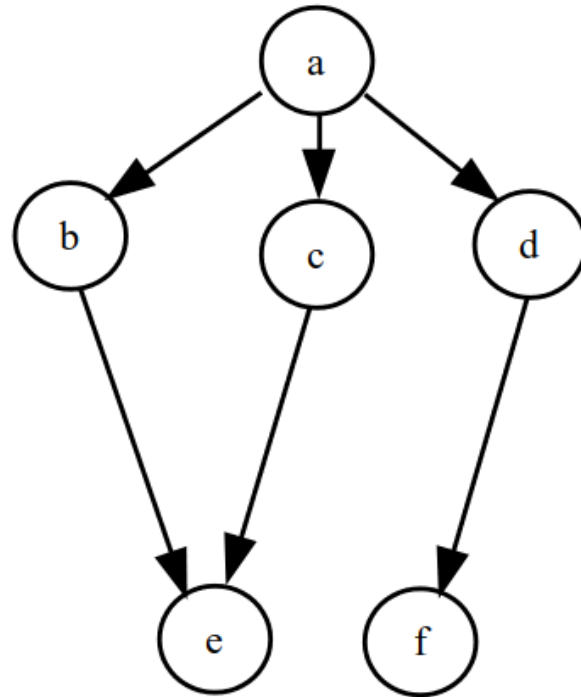
```
begin  
  wait(sc);  
  wait(sc);  
  c;  
  signal(se);  
end
```

begin

```
  initial(sc,1);  
  initial(sd,0);  
  initial(se,0);  
  cobegin  
    P1; P2; P3;  
  coend  
end.
```

Septiembre 2019

SOLUCIÓN



Febrero 2018

Usando semáforos, haga que, de forma cíclica, el proceso P1 acceda dos veces de forma consecutiva a la sección crítica, seguido por un acceso del proceso P2, es decir, la secuencia de entrada de la sección crítica deberá ser:
P1, P1, P2, P1, P1, P2, P1, P1, P2

Programa dosporuno

var

process P1

begin

repeat

Sección Crítica

Resto1

forever

end

process P2

begin

repeat

Sección Crítica

Resto2

forever

end

begin

cobegin

P1;P2;

coend

end

Febrero 2018

SOLUCIÓN.

Program dosporuno

var

S1, S2: SEMAPHORE;

process P1

begin

repeat

WAIT(S1);

Sección Crítica

SIGNAL(S2);

Resto1

forever

end

process P2

begin

repeat

WAIT(S2);

WAIT(S2);

Sección Crítica

SIGNAL(S1);

SIGNAL(S1);

Resto2

forever

end

begin

INITIAL(S1, 2);

INITIAL(S2, 0);

cobegin

P1; P2;

coend

end

Febrero 2019

Usando únicamente semáforos, haga que, de forma cíclica, los procesos accedan a la sección crítica en la siguiente secuencia:

P1, P2, P2, P3, P1, P2, P2, P3, P1, P2, P2, P3,

Program unodosuno

var

process P1

begin

repeat

Sección Crítica

Resto1

forever

end

process P2

begin

repeat

Sección Crítica

Resto1

forever

end

process P3

begin

repeat

Sección Crítica

Resto2

forever

end

begin

cobegin

P1;P2;P3;

coend

end

Febrero 2019

SOLUCIÓN.

Program unodosuno

var

S1,S2,S3:SEMAPHORE;

process P1

begin

repeat

 WAIT(S1);

 Sección Crítica

 SIGNAL(S2);

 SIGNAL(S2);

 Resto1

forever

end

Process P2

begin

repeat

 WAIT(S2);

 Sección Crítica

 SIGNAL(S3);

 Resto2

forever

end

Process P3

begin

repeat

 WAIT(S3);

 WAIT(S3);

 Sección Crítica

 SIGNAL(S1);

 Resto2

forever

end

begin

 INITIAL(S1,1);

 INITIAL(S2,0);

 INITIAL(S3,0);

 cobegin

 P1;P2;P3;

 coend

end

Septiembre 2019

Usando únicamente semáforos, haga que, de forma cíclica, los procesos accedan a la sección crítica en la siguiente secuencia:

P2, P3, P2, P3, P1 P2, P3, P2, P3, P1 P2, P3, P2, P3, P1

Program dtdtu

```
var
process P1
begin
  repeat
    Sección Crítica
    Resto1
  forever
end

process P2
begin
  repeat
    Sección Crítica
    Resto2
  forever
end

process P3
begin
  repeat
    Sección Crítica
    Resto3
  forever
end

begin
  cobegin
    P1;P2;P3;
  coend
end
```


Septiembre 2019

SOLUCIÓN 1

Program dtdu

var

s1,s2,s3:semaphore;

process P1

begin

repeat

wait(s1);

Sección Crítica

signal(s2);

Resto1

forever

end

process P2

begin

repeat

wait(s2);

Sección Crítica

signal(s3);

Resto1

forever

end

process P3

i: integer;

begin

i=0;

repeat

wait(s3)

Sección Crítica

i++;

if (i<2)

signal(s2);

else

signal(s1);

i=0;

Resto2

forever

end

begin

initial(s1,0);

initial(s2,1);

initial(s3,0);

cobegin

P1;P2;P3;

coend

end

Septiembre 2019

SOLUCIÓN 2

Program dtdu

var

s1,s2,s3:semaphore;

process P1

begin

repeat

wait(s1);

signal(s2);

wait(s1);

Sección Crítica

signal(s2);

Resto1

forever

end

process P2

begin

repeat

wait(s2);

Sección Crítica

signal(s3);

Resto1

forever

end

process P3

begin

repeat

wait(s3)

Sección Crítica

signal(s1);

Resto2

forever

end

begin

initial(s1,0);

initial(s2,1);

initial(s3,0);

cobegin

P1;P2;P3;

coend

end