

PARALEL PROGRAMLAMA ARASINAV YANITLAR

SORU 1:

Hayır, aşağıdaki senaryo gerçekleşirse, her iki proses de kritik bölüme ilerler:

Önce P1 çalışır 1.,2., ve 3. adımları yürütür ve kesilir—ilk while koşulu doğru bulmuş, ancak ikinci while false olduğundan (bekliyor [0]=false) çevrimden çıkmıştır.

Bu aşamada P0 çalışır, ilk while çevrimine girmez ($0 \neq 1$), 4. adımda sıra'ya 0 değerini atar ve kritik bölüme ilerler.

P1 çalışır, 4. Adımı yürütür, sıra'ya 1 değerini atar, 1. Adımdaki while'a döner ancak koşulu false bulur ($1 \neq 1$) ve çevrime girmeden 4. Adımdan devam eder, sıra'ya 1 değerini atar ve kritik bölüme ilerler.

SORU 3:

```
create_mailbox(m1);
create_mailbox(m2);
msg= "boşmesaj"
```

Proses A:

Begin

....

```
for (i=0; i<2; i++) send (m1, msg);
end;
```

Proses B:

begin

receive (m1);

....

```
send (m1, msg);
end;
```

Proses C:

begin

receive (m1);

....

```
send (m1, msg);
end;
```

Proses D:

begin

```
for (i=0; i<2; i++) receive (m2, msg);
```

....

End;

Asenkron haberleşme--- beklemez send, beklemeli receive

SORU 4:

```
int sayı=0; // sistemde aktif proses (arayıcı + ekleyici) toplam sayısı
sem sayı_dışla=1;
sem liste_dışla = 1
sem ekle_izni=1;
```

Proses Arayıcı:

```
{while (true) do {  
P(sayı_dışla);  
    sayı ++;  
    if (sayı == 1) P(liste_dışla); // sistemde ilk proses, erişim izni al  
V(sayı_dışla);
```

.... Arama işlemleri

```
P(sayı_dışla);  
    sayı --;  
    if (sayı == 0) V(liste_dışla); // son arayıcı prose  
V(sayı_dışla);  
};
```

Proses Ekleyici:

```
{while (true) do {  
P( ekle_izni=; // tek ekleyici proses olmalı, diğerleri dışla  
P(sayı_dışla);  
    sayı ++;  
    if (sayı == 1) P(liste_dışla); // sistemde ilk proses, erişim izni al  
V(sayı_dışla);
```

.... Ekleme işlemleri

```
P(sayı_dışla);  
    sayı --;  
    if (sayı == 0) V(liste_dışla); // son arayıcı prose  
V(sayı_dışla);  
V( ekle_dışla)  
};
```

Proses Silici:

```
{while (true) do {  
P(liste_dışla);
```

.... Silme işlemleri

```
V(liste_dışla);}  
};
```

SORU 5:**Monitor Yay ();**

```
bool aldım [1..n];  
bool ye ni_veri;  
int tüketici;  
data veri;  
cond yayıncı;  
cond alıcı[1..n];
```

entry yayınla (data v);

begin

if (yeni_veri) then c_wait (yayıncı); // bir önceki veri tüm alıcılara ulaşmamış, bekle

veri:= v; yeni_veri:= true;

for i:= 1 to n do

c_signal (alıcı [i]); // bekleyen alıcılar varsa uyandır

end;

entry veri_al (data veri, int i);

data t_veri;

begin

if (yeni_veri && (aldım[i]) then c_wait (alıcı [i]); // yeni veri yoksa bekle

t_veri:=veri;

aldım[i]:= true;

tüketici++;

if (tüketici == n) //son tüketici, veri tüm alıcılara ulaştı

then begin

yeni_veri := false;

for i:=1 to n do

aldım[i]:= false;

c_signal (yayıncı); //bekleyen yayıncı varsa uyandır

end;

end;

begin

for i:=1 to n do aldım[i]:= false; //ilk değerler

yeni_veri := false;

tüketici:=0;

end.