



Блокировка ПАМЯТИ

Выполнили: Орлов П.В.
Щукин М.А

Введение.

Все известные средства решения проблемы взаимного исключения основаны на использовании специально введенных аппаратных возможностей. Блокировка памяти относится к таким аппаратным возможностям.

Использование блокировки памяти при синхронизации параллельных процессов.

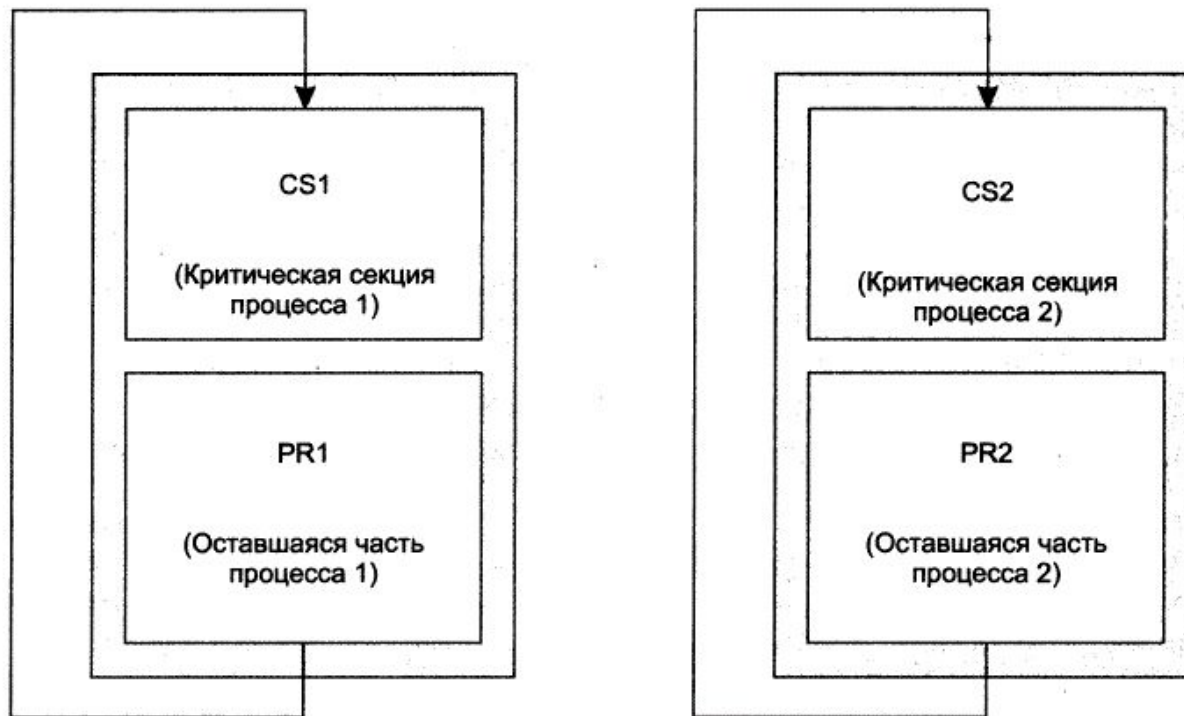
Блокировка памяти запрещает одновременное исполнение двух (и более) команд, которые обращаются к одной и той же ячейке памяти.

Это обязательное условие функционирования компьютера.

Заголовок слайда

Механизм блокировки памяти предотвращает одновременный доступ к разделяемой переменной, но не предотвращает чередование доступа. Таким образом, если критические секции исчерпываются одной командой обращения к памяти, данное средство может быть достаточным для непосредственной реализации взаимного исключения.

Возможные проблемы при организации взаимного исключения при условии использования только блокировки памяти.



Модель взаимодействующих процессов.

Листинг программы для первого решения.

```
Var перекл : integer;  
Begin перекл := 1; {при перекл=1 в критической секции находится процесс ПР1}  
Parbegin  
While true do  
Begin  
while перекл = 2 da begin end:  
CS1; { критическая секция процесса ПР1 }  
перекл := 2;  
PR1; { оставшаяся часть процесса ПР1 }  
End  
And  
While true do  
Begin  
while перекл = 1 do begin end;  
CS2; { критическая секция процесса ПР2 }  
перекл := 1;  
PR2; { оставшаяся часть процесса ПР2 }  
End  
Parend  
End.
```

Листинг второго варианта реализации взаимного исключения.

```
Var перекл1.перекл2.: boolean;
begin переМ1:=false;
переМ2:=false;
parbegin
while true do
begin
while перекл2 do
begin
end;
перекл1:=true;
CS1 { критическая секция процесса ПР1 }
перекл1:=false;
PR1 { процесс ПР1 после критической секции }
end
and
while true do
begin
while перекл1 do
begin
end;
перекл2:=true;
CS2 { Критическая секция процесса ПР2 }
перекл2:=false;
PR2 { процесс ПР2 после критической секции }
end
parend
end.
```

Листинг третьего варианта реализации взаимного исключения

```
var перекл1, перекл2 : boolean;
begin перекл1:=false; перекл2:=Ta15e;
parbegin
  ПР1: while true do
  begin
  перекл1:=true;
  while перекл2 do
  begin end;
  CS1 { критическая секция процесса ПР1 }
  перекл1:=Ta15e;
  ПР1 { ПР1 после критической секции }
  end
  and
  ПР2: while true do
  begin
  перекл2:=true;
  while перекл1 do
  begin end;
  CS2 { критическая секция процесса ПР2 }
  перекл2:=false;
  ПР2 { ПР2 после критической секции }
  end
parend
end.
```