سپهر مقیسه

9831103

تکلیف 8

1. void acquire(lock \*mutex) {  
   while (compare and swap(&mutex->available, 0, 1) != 0)  
   ;  
   return;  
   }

void release(lock \*mutex) {  
mutex->available = 0;  
return;  
}

2-در اول هر دو p0-p2 در wait میمانند و p1 یک بار عبارت را چاپ کرده و سپس دو فرایند دیگر را بیدار میکند و این دو فرایند دو بار دیگر سمافور p1 را افزایش میدهند و بعد از آن تمام میشوند و فرایند p1 بعد از دو بار دیگر چاپ کردن در wait p1 گیر میکند .پس در مجموع سه بار

3-الف)test\_&\_set(int \* lock){

Return compare\_&\_swap(lock,0,1);

}

ب)در لحظه خواندن داده ای که مشترک است نیاز است atomic باشد و در تمامی لحظات مهم نیست

ج)چرا که تک هسته فقط یک فرایند را انجام میدهد .در آن زمان خواندن lock و سپس قفل دوباره آن است و در این حین فرآیند دیگری وارد نمیشود.

د)خیر چرا که قفل در یک زمان در دست یکی از فرایند ها است فقط و دیگری نمیتواند وارد شود.با استفاده از توابعی مانند dpram میتوان دستور های test and set را کنترل کرد