

گزارش پروژه 4 آزمایشگاه

سیدعلیرضا میرشفیعی - 810101532

امیرحسین صفری - 810101571

محمد صدرا عباسی - 810101469

1- علت غیرفعال کردن وقفه این است که ممکن است وقتی که ما در حال استفاده از یک قفل هستیم، وقفه ای رخ دهد که نیازمند گرفتن همان قفل است. از آنجا که ما از قبل قفل را گرفته بودیم، interrupt handler منتظر می ماند تا قفل آزاد شود ولی این امکان وجود ندارد چرا که از قبل برنامه ما قفل را گرفته است. در این حالت deadlock رخ می دهد.

از `pushcli()` برای این استفاده می کنیم که هر وقت که یک قفل فعال می شود، این تابع وقفه ها را غیر فعال می کند و مقدار `ncli` را افزایش می دهد. از تابع `popcli()` هم زمانی استفاده می کنیم که می خواهیم قفلی را رها کنیم. هر بار که این تابع فراخوانی شود، یک واحد

از `ncli` کم می کنیم و در صورتی که مقدار `ncli` برابر صفر شده بود،  
وقفه ها را فعال می کنیم.

توابع `cli()` و `ncli()` مسئولیت فعال سازی و غیر فعال سازی قفل را  
دارند ولی تعداد دفعات دریافت و رها سازی قفل تا اینجا ندارند. توابع  
`pushcli()` و `popcli()` برای فعال سازی و غیر فعال سازی قفل ها از  
این توابع استفاده می کنند.

-2

**UNUSED:** هنوز به پردازش حافظه ای تخصیص نیافته است.

**EMBRYO:** در این حالت پردازش در یک مرحله انتقالی برای مقداردهی

اولیه قرار دارد. برای مثال در این حالت `struct proc`

اختصاص داده شده است اما `user address space`

اختصاص داده نشده است.

**SLEEPING:** پردازش منتظر یک `event` مانند انجام عمل `IO` است.

**RUNNABLE:** پردازش آماده اجرا است و منتظر است تا `scheduler`

یک `cpu` را به این پردازش اختصاص دهد.

**RUNNING**: پردازش در حال اجرا بر روی یک **cpu** است.

**ZOMBIE**: پردازش کارش را تمام کرده و منتظر **parent** است تا **wait**

را صدا کرده و برخی منابع مانند اطلاعاتی در **ptable** را آزاد کند.

تابع **sched** کنترل **cpu** را از پردازش ای که در آن اجرا میشد گرفته و این کنترل را به **scheduler** می دهد تا پردازش بعدی را برای اجرا بر روی **cpu** انتخاب کند.

-3

مزایا:

1- با استفاده از **reader-writer lock** این امکان وجود

دارد که چندین **reader** بتوانند به یک داده مشترک

برای خواندن دسترسی داشته باشند. در حالی که در

**ticket lock** هر پردازش یا نخ باید منتظر بماند تا یکی

یکی به این داده مشترک برای خواندن دسترسی پیدا

کنند. این مزیت بیشتر خود را در مواردی که تعداد زیادی reader در سیستم وجود دارند نشان می دهد. بدلیل امکان دسترسی همزمان چند reader به داده مشترک و باعث عدم وجود race condition می شود.

2- اولویت دهی به writer در حالت صورت مسئله سبب می شود که writer دچار starvation نشود. اما در ticket lock اگر تعداد پردازنده ها و نخ ها بسیار زیاد باشد، یک writer باید مدت زمان نسبتاً زیادی را منتظر بماند تا بتواند قفل را بگیرد و کارش را انجام دهد.