



بسمه تعالی  
سیستم‌های عامل  
نیمسال اول ۹۹-۹۸  
تمرین (۰۸)



دانشکده مهندسی کامپیوتر

مهلت تحویل: ۱۳۹۸/۰۹/۱۰

دانشگاه صنعتی امیرکبیر

شماره دانشجویی: ۹۶۳۱۰۰۱

نام و نام خانوادگی: محمدرضا اخگری

۱. busy waiting عملی که به یک موضوع یا پردازش اجازه می دهد تا در حالی که منتظر چیزی هستید از زمان CPU به طور مداوم استفاده کنید. به معنای این است که یک فرآیند در یک حلقه منتظر است و در هنگامی که نوبتش میشود (برای مثال فرآیند دیگری وارد ناحیه بحرانی شده بود و الان خارج شده و نوبت این فرآیند است) از آن خارج میشود.

انتظارهای دیگر: یک فرآیند در انتظار وقوع یک رویداد در صف انتظار است (مثلاً I / O semaphore) و بدون آنکه CPU به آن اختصاص داده شود ، این کار را انجام می دهد.

میتوان از busy waiting چشم پوشی کرد، اگرچه این کار باعث تحمل کردن سربار میشود بدین طریق که فرآیند را به حالت خواب ببریم و زمانی که نوبتش میشود بیدارش کنیم.

۲. عملیات wait مقدار یک متغیر را کاهش میدهد. اگر این عمل به صورت اتمیک اجرا نشود فرض کنیم حالت زیر پیش میاید:

متغیر ما مقدار صفر دارد، فرآیند اول wait را فراخوانی میکند. فرآیند دو هم wait را فراخوانی میکند. هر دو مقدار صفر را میبینند و در نهایت منفی یک مقدار متغیر میشود ولی هردو فرآیند وارد ناحیه بحرانی شده اند.

۳. الف) امکان پیش آمدن بن بست در این مدل نیست.

ب) امکان پیش آمدن قحطی در این مدل هست. بدین طریق که هر موقع که قرار به رسیدن به نوبت به این فرآیند بشود. فرآیند دیگری زودتر وارد شود. ( با توجه به اینکه اولییتی برای ورود به تابع نداده است)



بسمه تعالی  
سیستم‌های عامل  
نیمسال اول ۹۹-۹۸  
تمرین (۰۸)



دانشکده مهندسی کامپیوتر

مهلت تحویل: ۱۳۹۸/۰۹/۱۰

دانشگاه صنعتی امیرکبیر

شماره دانشجویی: ۹۶۳۱۰۰۱

نام و نام خانوادگی: محمدرضا اخگری

۴.

الف) یک خواننده میتواند از `await(active_writers == 0 && waiting_writers == 0)` استفاده کند، برای چک کردن اینکه آیا نویسنده ای الان در حال نوشتن است یا نویسنده ای منتظر ورود به ناحیه بحرانی است یا خیر؟ در صورت وجود شرایط وارد ناحیه بحرانی میشود.

یک نویسنده هم میتواند از `await(active_readers == 0 && active_writers == 0)` استفاده کند تا مطمئن شود نویسنده دیگری در حال نوشتن نباشد یا خواننده ای در حال خواندن نباشد.

ب) سیستم مجبور خواهد بود تا چک کند کدام یک از تردهایی که در حالت `wait` هستن برای مرحله بعدی باید انتخاب شوند (با چک کردن شرط هایی که در قسمت الف تعریف شده است). این پیچیدگی زیادی دارد و ممکن است نیاز به تعامل با کامپایلر داشته باشد.

ج) به چرت و پرتیایی که پایین نوشتم کاری نداشته باشید، فقط گذاشتم که بعدا داشته باشمش.

One could restrict the Boolean condition to be a disjunction of conjunctions with each component being a simple check (equality or inequality with respect to a static value) on a program variable. In that case, the Boolean condition could be communicated to the run-time system, which could perform the check every time it needs to determine which thread to be awakened.