تمرین ۳

دو گزینه برای پیاده سازی mutex و سمافور شمارشی وجود دارد

راه اول سمافور قوی میباشد؛ سمافور قوی از صف استفاده میکند و هنگامی که فرآیندها برای وارد شدن به ناحیه بحرانی wait میشوند؛ مطابق با ترتیب wait شدنشان وارد ناحیه بحرانی می‌شوند.

راه دوم سمافور ضعیف میباشد؛ سمافور ضعیف به این صورت میباشد که فرآیندهایی که به wait میخورند را نگه میدارد و هنگامی که یک فرآیند signal بزند؛ به صورت تصادفی یکی از فرآیندهایی که wait شده بود را انتخاب میکند.

مزیت اصلی سمافور قوی به ضعیف؛ سهولت استفاده و ضمنا استفاده بهتر از حافظه است.

تمرین ٤

انواع موازی سازی :

: data parallelism

برای نگهداری اطلاعات مشترک در بین هسته‌های مختلف cpu است؛ به این صورت که فرض کنیم که میخواهیم تعدادی عدد را جمع کنیم؛ نیمی از این اعداد را یک هسته cpu و نیم دیگر را هستهای دیگر محاسبه میکند. و در نهایت به کمک سیاست موازی سازی data parallelism مجموع کل را محاسبه میکنیم.

: task parallelism

در این روش، سیاست گذاری بر روی به اشتراک گذاشتن taskهاست؛ به این شکل که هر نخ میتواند کار متفاوتی را انجام دهد و درواقع هر نخ کار متفاوتی را انجام میدهد. و در این حالت این احتمال وجود دارد که نخ‌ها به اطلاعات مشترک یا متفاوتی نیاز داشته باشند که دچار مشکلی نمیشود.

تمرین ٥

به علت نبود اینترنت دستمان به جایی بند نبود 😊

تمرین ٨

Memory transaction

به تعدادی از دستورات خواندن و نوشتن پیاپی گفته میشود که بایستی atomic اجرا شوند(منظور از atomic این است که یا کلیه آن‌ها انجام شوند یا هیچ کدام انجام نشوند) چنانچه همه این دستورات انجام شوند آن کار به طور کامل انجام شده است و آن سری دستورات commit میشوند. و در غیر این صورت این دستورات میبایستی از ابتدا اجرا شوند (و تغییرات آن‌ها اعمال نمیشود) به این اتفاق rollback میگویند.

گاهی وقت‌ها به خاطر استفاده پردازش‌های چند هستهای ممکن است به علت وجود نواحی بحرانی دچار اشکالاتی نظیر dead-locking و مشکلاتی نظیر آن شویم. به همین علت میتوانیم ناحیه بحرانی خود را به صورت atomic تعریف کنیم که یا به طور کامل انجام شود و یا اصلا انجام نشود.

تمرین ۱۲

مانیتور دارای سه ویژگی میباشد:

ویژگی اول : اطلاعات محلی موجود در سیستم مانیتور به هیچ وجه توسط فرآیندهای بیرونی قابل دسترسی نیست و فقط فرآیندهای موجود در همان سیستم امکان دسترسی به آن را دارند.

ویژگی دوم : یک فرآیند تنها زمانی وارد مانیتور میشود که یکی از فرآیندها را موجود در آن را بیدار کند.

ویژگی سوم : اگه در لحظه به مانیتور نگاه کنیم میبینیم که فقط یکی از فرآیندها در حال اجرا است :

چون فقط یکی از فرآیندها در لحظه اجرا میشوند میتوان نواحی بحرانی را در آن قرار داد تا در لحظه فقط یکی از فرآیندها بتواند وارد ناحیه بحرانی شود. دستوراتی برای wait و سیگنال در زبان C وجود دارد که به مانیتور کردن کمک میکند مثل : cwait() و csignal()

تمرین ۱۳

همانطور که میدانیم برای پیاده سازی synchronization میبایستی فرآیندها به فازهای مختلف تقسیم میشوند و فرآیندهای فاز بعدی نمیتوانند شروع شوند مگر این که فرآیندهای فاز قبلی به طور کامل پروسس شده باشند. برای پیاده سازی این موضوع از barrier استفاده میشود. به این صورت که چنانچه همه فرآیندهای یک فاز به پایان نرسیده باشد؛ فرآیندهای به پایان رسیده از همان فاز میبایستی صبر کنند تا دیگر فرآیندها نیز تمام شوند (در پایان هر فاز بایستی از barrier استفاده کرد)