پروژه چهارم آزمایشگاه سیستم عامل

مرتضی نوری (810198481)
شایان شاه محمدی (810198531)
سید محمد امین اطیابی (810198559)

۱.

چون وقفه ها بالاترین اولویت را دارند، در هر لحظه ممکن است کد هسته متوقف شود تا interrupt handler اجرا شود. بنابرین برای محافظت از ناحیه بحرانی و همچنین برای جلوگیری رخ دادن deadlock نیاز است که وقفه ها غیرفعال شوند.(برای مثال عدم غیر فعال کردن وقفه ها باعث ایجاد deadlock در فرایند اجرای iderw و ideintr شود.)

در xv6 برای مدیریت ناحیه های بحرانی تودرتو از توابع pushcli (برای غیر فعال کردن وقفه ها) و popcli(برای فعال کردن وقفه ها) استفاده می شود. این دوتابع از توابع cli و sti استفاده می کنند تا وقفه ها را غیر فعال یا فعال کنند علاوه بر آن، این دو تابع قابلیت های اضافه دیگری نیز دارند: اورهایی که رخ داده است چاپ می کنند ، از متغیر های ncli و intena که در داده ساختار وضعیت پردازنده هستند استفاده می کنند تا ناحیه های بحرانی تودرتو را مدیریت کنند و اطمینان حاصل کنند که همه قفل ها آزاد شوند و سپس وقفه ها غیر فعال شوند.

۲.

در تابع acquiresleep، تا وقتی که پردازه فرصتی برای به دست گرفتن قفل نداشته باشد، با استفاده از تابع sleep به خواب می رود در نتیجه busy waiting نداریم.

هنگامی که تابع releasesleep صدا زده می شود، قفل آزاد شده و تمام پردازه های روی کانال قفل بوسیله wakeup از حالت SLEEP به حالت RUNNABLE می روند.

در مساله producer-consumer غالبا مدت زمان انتظار برای آزاد شدن قفل زیاد است و به همین دلیل busy waiting که در قفل های چرخشی وجود دارد باعث افت شدید کارایی سیستم خواهد شد.

۳.

UNUSED : پردازه ای که تا الان هیچ استفاده ای از آن نشده است.(منابع، cpu به آن اختصاص داده نشده است.)

EMBYRO: وقتی allocproc صدا زده شد پردازه ای که unused بوده است به این استیت تغییر می کند یعنی پردازه می تواند ادامه مراحل را طی کند تا به پایان برسد.

SLEEPING: منابع مورد نیاز پردازه تامین نشده است.(scheduler از این پردازه اسفاده نمی کند.)

RUNNABLE: پردازه ای که همه منابع برای اجرا را در اختیار دارد و منتظر است که cpu به آن اختصاص داده شود.

RUNNING: پردازه ای که به آن cpu اختصاص داده شده و در حال اجراست.

ZOMBIE: پردازه ای که کارش تمام شده است اما اطلاعات آن هنوز در ptable موجود است.(این حالت وقتی اتفاق می افتد که پردازه پدر wait را صدا نکرده باشد.)

برای زمان بندی پردازه جدید در ابتدا تابع() sched صدا زده می شود، در این تابع بعد از چک کردن خطاهایی که ممکن است رخ دهد (وقفه ها فعال باشند، پردازه در حالت اجرا باشد، قفل ptable گرفته نشده باشد و ...) در صورت عدم وجود خطا، عملیات context switch را بین پردازه فعلی و پردازه حاضر در scheduler پردازنده را انجام می دهد.

در واقع هرجا نیاز باشد که زمانبندی صورت بگیرد از تابع sched استفاده می شود. برای مثال در توابع exit و yield .

۴.

در لینوکس mutex به این صورت است که در ساختار آن owner تعریف شده است که مشخص کند کدام پردازه قفل را در اختیار دارد. این قفل ویژگی های زیر را دارد:

- تنها نگهدارنده قفل می تواند آن را رها کند.
- رها سازی قفل های متعدد در یک زمان مجاز نیست.
- انتظار مشغول شده نداریم

لینک زیر تعریف mutex در لینوکس را نشان می دهد:

https://github.com/torvalds/linux/blob/master/include/linux/mutex.h

```
void
releasesleep(struct sleeplock *lk)
{
  if(myproc()->pid == lk->pid)
  {
    acquire(&lk->lk);
    lk->locked = 0;
    lk->pid = 0;
    wakeup(lk);
    release(&lk->lk);
  }
}
```

۵.

مفهوم حافظه تراکنشی در دیتابیس ها مطرح می شود که با الهام از آن می توان همگام سازی را در سطح نرم افزار و سخت افزار انجام داد.

تراکنش حافظه (memory transaction) : دنباله ای از عملیات های خواندن و نوشتن در حافظه است که به صورت atomic انجام می شوند. اگر تمام عملیات ها در تراکنش با موفقیت به اتمام برسند، تراکنش حافظه ای ثبت می شود. (commit) در غیر اینصورت عملیات متوقف می شود و برگشت داده می شود.(rolled back)

این روش کار برنامه نویسان را برای همگام سازی راحت می کند، زیرا کافی است که مثلا ساختاری مانند{ S} atomic به زبان اضافه شود به گونه ای اطمینان حاصل شود که عملیات S مانند یک تراکنش اجرا شود. به این ترتیب اطمینان حاصل می شود که ناحیه بحرانی محافظت می شود.

فواید:

- وظیفه atomic کردن عملیات ها از عهده برنامه نویس خارج می شود.

- چون از قفل استفاده نمی شود، بنابراین deadlock هم نخواهیم داشت.
- با افزایش ریسه ها، مدیریت همگام سازی با استفاده از قفل های سنتی مانند mutex و semaphore دشوار خواهد شد زیرا سربار و اختلاف ریسه ها برای نگهداری قفل (lock ownership) بسیار زیاد خواهد شد.

نمونه اجرا برنامه فلاسفه خورنده