



اگر چه موفقیت سیستم عامل بهسید (بهترین سیستم عامل دنیا) فراتر از انتظار شرکت خوخو بوده است، خوخو قصد دارد نقش پررنگ‌تری در آینده‌ی دنیای سیستم عامل‌ها بازی کند. شایعاتی به گوش می‌رسد که این شرکت قصد دارد سیستم عامل جدیدی را به نام نگنو (نگاه نو) طراحی کند. برخی از برنامه‌نویسان خوخو در حال پاسخ به سؤالاتی هستند که به نظر می‌رسند در ارتباط با این سیستم عامل جدید باشند. به آنها کمک کنید.

- ۱- فرض کنید برای سادگی سیستم عامل و سخت‌افزارهای مورد نیاز، همه‌ی قسمت‌های یک سیستم عامل و پردازنده‌های کاربری فقط در یک حالت پردازنده (حالت بدون محدودیت) اجرا شوند. این کار چه بدی‌هایی دارد؟
- ۲- برای کاهش هزینه‌های رخدادهای طبیعی، خوخو یک دستگاه ورودی و خروجی را به کامپیوترهایش اضافه کرده است. این دستگاه حسگری برای تشخیص لرزش زمین دارد و رخداد لرزش را به سیستم عامل گزارش می‌دهد. قرار است سیستم عامل، پس از تشخیص لرزش، سیگنال جدید SIGEQUAKE را برای همه‌ی پردازنده‌ها بفرستد. توضیح دهید سیستم عامل با چه روش‌هایی می‌تواند از رخداد لرزش زمین آگاه شود.
- ۳- فرض کنید A سیستم عاملی باشد که در یک ماشین مجازی در حال اجرا است و B سیستم عامل میزبان آن باشد. آیا یک عمل ورودی یا خروجی در پردازنده‌ی در A موجب یک عمل ورودی یا خروجی در B می‌شود؟ اگر خیر، چرا و اگر بله، در چه صورتی؟
- ۴- با بیان دلیل، درستی این گزاره را مشخص کنید: در تبادل پیغام، اگر دریافت Blocking باشد، ارسال هم قطعاً Blocking است.
- ۵- با بیان دلیل، درستی این گزاره را مشخص کنید: هر چه حجم اطلاعاتی که توسط یک لوله انتقال می‌یابد بیشتر باشد، اندازه‌ی لوله نیز بیشتر می‌شود.
- ۶- بندهای یک پردازنده، آن پردازنده را در یکی از دو وضعیت دریافت و ارسال قرار می‌دهند. برای دریافت، یک بند تابع lock\_recv و در پایان دریافت تابع unlock\_recv را فراخوانی می‌کند. برای ارسال، یک بند تابع lock\_send و در پایان ارسال تابع unlock\_send را فراخوانی می‌کند. این چهار تابع را به کمک سمافور یا مانیتور صورتی پیاده‌سازی کنید که اگر پردازنده در وضعیت دریافت قرار داشته باشد، بندهایی که قصد ارسال دارند، منتظر شوند تا آخرین پردازنده‌ی دریافت‌کننده تابع unlock\_recv را فراخوانی کند. به صورت مشابه، اگر پردازنده در وضعیت ارسال قرار داشته باشد، پردازنده‌هایی که قصد دریافت دارند منتظر می‌شوند تا آخرین پردازنده‌ی ارسال‌کننده تابع unlock\_send را فراخوانی کند.



۷- آیا در مسئله‌ی قبل، امکان گرسنگی وجود دارد؟ اگر بله، در چه حالت یا حالت‌هایی رخ می‌دهد؟

۸- در قطعه کد زیر، sem و val در حافظه‌ی مشترک هستند و دو بند تابع work را فراخوانی می‌کنند. بند اول این تابع را با ورودی یک و بند دوم این تابع را با ورودی دو فراخوانی می‌کند. پس از توقف اجرای دو بند، همه‌ی مقدارهای ممکن برای متغیر val را مشخص کنید؛ برای هر یک از این مقادیر ترتیب اجرای خطوط کد توسط دو تابع را مشخص کنید.

```

1 semaphore sem = 0;
2 int val = 2;
3 void work(int id)
4 {
5     if (id == 1) {
6         wait(sem);
7         val = val * 3;
8         signal(sem);
9     } else {
10        val = val + 1;
11        signal(sem);
12    }
13    if (id == 2) {
14        wait(sem);
15        val = val + 1;
16    }
17 }
```

۹- در قطعه کد زیر دو آرایه‌ی تعریف شده در حافظه‌ی مشترک قرار دارند و تابع transact توسط تعدادی بند فراخوانی می‌شود؛ عددی که به عنوان ورودی به این تابع فرستاده می‌شود از صفر تا هشت است. با فرض اینکه مقدار اولیه‌ی سمافورهای آرایه‌ی sems یک باشد، حداقل چند بند لازم است تا بن بست رخ دهد؟ اگر بن بست رخ می‌دهد، برای هر بند نشان دهید که با چه ورودی تابع را فراخوانی کرده است و در حال اجرای چه خطی است.

```

1 semaphore sems[9];
2 int account[9];
3 void transact(int i)
4 {
5     int j = (i + 3) mod 9;
6     wait(sems[i]);
7     wait(sems[j]);
8     account[i] -= 5;
9     account[j] += 5;
10    signal(sems[i]);
11    signal(sems[j]);
12 }
```