

به نام خدا

امتحان آزمایشگاه سیستم عامل

ترم ۱-۹۹-۹۸، گروه چهارشنبه صبح

از بین دو سؤال زیر، یک سؤال را با اختیار خود انتخاب کنید.

لطفاً همه کدهای جواب خود را در فولدری با نام stdno_wed1 ذخیره کرده فشرده‌سازی کنید و تحویل TA دهید

بازی نیم بدین صورت اجرا می‌شود: یک کیسه حاوی تعدادی سنگ وجود دارد. هر بازیکن در نوبت خود یک تعداد تصادفی از سنگ‌ها را از کیسه برمی‌دارد. بازیکنی که آخرین سنگ‌ها را برداشت به طوری که کیسه خالی شود، برنده خواهد بود. بازی به نوبت انجام نمی‌شود و هر بازیکن در فواصل زمانی مختلف ممکن است بازی کند. در این بازی، غیر از بازیکن‌ها یک مدیر جهت مدیریت و یا ارتباط بین بازیکن‌ها حضور دارد.

با استفاده از فقط یکی از روش‌های زیر بازی را به صورتی که بیان شد، شبیه‌سازی کنید.

سؤال اول (برنامه‌نویسی شبکه و چندنخی)

- یک برنامه‌ی سرور TCP برای مدیریت بازی و ارتباط بین بازیکن‌ها بنویسید: این سرور تعداد بازیکن‌ها (کلاینت‌ها) و تعداد سنگ‌های موجود کیسه را از طریق آرگومان ورودی دریافت کرده، سپس صبر می‌کند تا همه‌ی بازیکن‌ها به او متصل شوند.
- هر بازیکن هر بار بخواهد یک مرحله بازی کند، به صورت تصادفی عددی کمتر از ۵ (مثلاً عدد T) انتخاب کرده و برای سرور می‌فرستد.

- هر بازیکن پس از یک مرحله بازی کردن، به اندازه T ثانیه (همان مقدار T در بند قبل) صبر می‌کند و سپس دوباره بازی می‌کند.

-- مدیر هر بار که مقداری از یک بازیکن دریافت می‌کند، مقدار num را آپدیت کرده (عدد دریافتی را از num کم می‌کند و اگر num منفی شد آن را صفر می‌کند) و پیامی به شکل زیر چاپ می‌کند:

```
player port_num plays, num= val
```

port_num شماره پورت بازیکن مربوطه و val مقدار جدید num است.

- هرگاه مقدار num صفر شد، آخرین بازیکنی که توانسته بازی کند و باعث صفر شدن num شده برنده است. در این صورت، مدیر اتصال همه بازیکن‌ها را بسته و پیام زیر را چاپ می‌کند. سپس به اجرای خود نیز خاتمه می‌دهد. همچنین بازیکن‌ها نیز به اجرای خود پایان می‌دهند.

```
player port_num is the winner
```

port_num شماره پورت بازیکن برنده است.

- برنامه اسکریپتی با نام game.sh بنویسید که با دریافت <تعداد بازیکن‌ها>، کدها را کامپایل کند و با آرگومان مناسب اجرا کند.

سؤال دوم (برنامه‌نویسی مالتی پروسس و pipe)

- بازی نیم را با کمک برنامه‌نویسی multiprocess بنویسید. در این بازی پروسس والد، مدیر و فرزندهای او همه بازیکن هستند.
- برنامه والد، تعداد بازیکن‌ها (پروسس‌های فرزند) و تعداد سنگ‌های موجود کیسه را از طریق آرگومان ورودی دریافت کرده، سپس همه پروسس‌های بازیکن را ایجاد می‌کند.

- بین پروسس‌ها یک pipe ساخته می‌شود. یک pipe دیگر نیز جهت اعلام برنده ساخته می‌شود.

- هر بازیکن هربار بخواهد یک مرحله بازی کند، عدد num را از pipe می‌خواند. سپس عددی تصادفی کمتر از ۵ (مثلاً T) انتخاب کرده از num کم می‌کند و num را دوباره روی pipe می‌گذارد (اگر num منفی شد، عدد صفر را روی پایپ می‌گذارد) و عبارت زیر را چاپ می‌کند:

```
player pid plays, num= val
```

pid همان pid پروسس بازیکن مربوطه و val مقدار جدید num است.

- هر بازیکن پس از یک مرحله بازی کردن، به مقدار T ثانیه (همان مقدار T در بند قبل) صبر می‌کند و سپس دوباره بازی می‌کند.

- اگر بعد از بازی یک بازیکن، مقدار num صفر شد، بازیکن موردنظر شماره pid خود را در pipe برنده می‌گذارد و یک سیگنال برای پروسس مدیر می‌فرستد.

- اگر مدیر متوجه اتمام بازی شد، شماره pid برنده را به صورت زیر چاپ می‌کند و با ارسال سیگنال به اجرای پروسس همه بازیکن‌ها خاتمه می‌دهد. همچنین به اجرای خود نیز پایان می‌دهد.

```
player pid is the winner
```

pid همان شناسه بازیکن برنده است.

- برنامه اسکریپتی با نام game.sh بنویسید که با دریافت <تعداد بازیکن‌ها>، کدها را کامپایل کند و با آرگومان مناسب اجرا کند.