

## گزارش پروژه‌ی درس اصول سیستم‌های عامل

علی علی محمدی

۹۶۱۳۰۲۷

این پروژه شامل دو قسمت است که یکی مربوط به برنامه‌نویسی سوکت و دیگری مربوط به برنامه‌نویسی حافظه‌ی مشترک است: در قسمت برنامه‌نویسی سوکت، باید ابتدا سرور را راه‌اندازی کنیم و سپس با کلاینت‌های متفاوت به آن متصل شویم. برای راه‌اندازی سرور از دستور زیر استفاده می‌کنیم:

```
$ ./server -h listenaddress -p portnumber
```

سپس از دستور زیر برای راه‌اندازی کلاینت‌ها استفاده می‌کنیم:

```
$ ./client -h serveraddress -p portnumber text
```

برای این که بتوانیم تعدادی کلاینت را به صورت همزمان به اجرا در بیاوریم و برای اندازه‌های مختلف پیام‌ها، تعدادی فایل اسکریپت شل (Shell Script) نوشته شده است که کافی است پس از اجرای سرور، آن‌ها را به اجرا در بیاوریم. (در این فایل‌ها تعداد کلاینت‌ها را برابر ۱۰ در نظر گرفته‌ایم که به دلخواه قابل تغییر است.) به عنوان مثال، برای ارسال پیام‌های ۱۰۰ بیتی می‌توان از دستور زیر استفاده کرد:

```
$ sh run_clients_100B.sh
```

به ازای هر ارتباط جدید، برنامه‌ی سرور، تعداد کلاینت‌هایی که را که تا کنون به آن متصل شده‌اند، نمایش خواهد داد.

نتایج تأخیر زمانی (بر حسب میلی ثانیه) را در جداول زیر می‌بینید:

اندازه‌ی پیام شماره‌ی کلاینت	۱۰۰ بایت	۱ کیلوبایت	۲ کیلوبایت	۴ کیلوبایت	۸ کیلوبایت
۱	0.024	0.028	0.028	0.040	0.055
۲	0.043	0.042	0.033	0.028	0.025
۳	0.036	0.021	0.027	0.047	0.034
۴	0.018	0.024	0.017	0.028	0.023
۵	0.023	0.017	0.016	0.025	0.017
۶	0.024	0.016	0.024	0.029	0.029
۷	0.019	0.020	0.016	0.029	0.034
۸	0.025	0.019	0.025	0.032	0.033
۹	0.019	0.038	0.044	0.032	0.019
۱۰	0.028	0.026	0.041	0.044	0.023
میانگین	0.026	0.027	0.0345	0.042	0.039

در قسمت برنامه‌نویسی حافظه‌ی اشتراکی، باید ابتدا سرور را راه‌اندازی کنیم و سپس با کلاینت‌های متفاوت به آن متصل شویم. برای راه‌اندازی سرور از دستور زیر استفاده می‌کنیم:

```
$ ./server
```

سپس از دستور زیر برای راه‌اندازی کلاینت‌ها استفاده می‌کنیم:

```
$ ./client text
```

برای این که بتوانیم تعدادی کلاینت را به صورت همزمان به اجرا در بیاوریم و برای اندازه‌های مختلف پیام‌ها، تعدادی فایل اسکریپت (Shell Script) نوشته شده است که کافی است پس از اجرای سرور، آن‌ها را به اجرا در بیاوریم. (در این فایل‌ها تعداد کلاینت‌ها را برابر ۱۰ در نظر گرفته‌ایم که به دلخواه قابل تغییر است.) به عنوان مثال، برای ارسال پیام‌های ۱ کیلوبایتی می‌توان از دستور زیر استفاده کرد:

```
$ sh run_clients_1KB.sh
```

نتایج تأخیر زمانی (بر حسب میلی ثانیه) را در جداول زیر می‌بینید:

اندازه‌ی پیام شماره‌ی کلاینت	۱۰۰ بایت	۱ کیلوبایت	۲ کیلوبایت	۴ کیلوبایت	۸ کیلوبایت
۱	0.025	0.029	0.021	0.054	0.030
۲	0.015	0.031	0.024	0.037	0.022
۳	0.030	0.022	0.017	0.034	0.026
۴	0.015	0.018	0.017	0.021	0.053
۵	0.015	0.012	0.022	0.023	0.034
۶	0.014	0.012	0.018	0.023	0.038
۷	0.024	0.015	0.011	0.033	0.029
۸	0.023	0.011	0.012	0.017	0.033
۹	0.012	0.012	0.015	0.052	0.027
۱۰	0.013	0.014	0.011	0.067	0.033
میانگین	0.019	0.0215	0.016	0.0605	0.0315

با مقایسه‌ی نتایج حاصل شده از دو قسمت با یکدیگر، به نظر می‌رسد که با توجه به بار پردازشی سیستم در زمان اجرای برنامه، تأخیرها به صورت یکنواخت با افزایش اندازه‌ی پیام، افزایش نمی‌یابند. با این حال، می‌توان استدلال‌هایی کلی ارائه داد؛ به عنوان مثال، می‌توان گفت که در اندازه‌های کوچک، حافظه‌ی اشتراکی، اندکی سریع‌تر از سوکت است. از طرفی مشاهده می‌شود که سوکت عملکرد پایدارتری دارد و به ازای اندازه‌های متفاوت، تغییرات در تأخیر آن، بسیار کم است و همچنین نسبت به تغییر بار پردازشی سیستم در لحظات مختلف، حساس نیست؛ این در حالی است که حافظه‌ی مشترک نسبت به تغییر بار پردازشی سیستم در لحظات مختلف، بسیار حساس است (مثلاً میانگین تأخیر پیام ۴ کیلوبایتی، دو برابر میانگین تأخیر پیام ۸ کیلوبایتی است که عجیب به نظر می‌رسد اما با توجه به بار پردازشی سیستم در زمان اجرا، قابل توجیه است)؛ زیرا وابستگی بیشتری به قسمت مشخصی از حافظه دارد و به قسمتی از حافظه محدود شده است. در مقابل، سوکت‌ها در استفاده از حافظه، آزادی عمل بیشتری دارند که باعث کاهش تأثیر وضعیت بار پردازشی لحظه‌ای سیستم بر عملکرد آن‌ها می‌شود. علاوه بر استدلال‌های فوق، می‌توان نتیجه گرفت که با افزایش اندازه‌ی پیام، تأخیر سوکت‌ها اندکی بیشتر می‌شود اما نمی‌توان با اطمینان نسبت به آن اظهار نظر کرد.