

پروژه‌ی کوچک اول: برنامه‌نویسی سوکت

می‌خواهیم با استفاده از برنامه‌نویسی سوکت، یک سیستم اشتراک فایل بسازیم. در این سیستم، اشتراک فایل‌ها میان کاربران به صورت P2P صورت می‌پذیرد. عمل‌کرد سیستم بدین‌صورت می‌باشد که یک سرور مرکزی در راس آن وجود دارد که هر یک از کاربران، فایل‌های خود را در آن سرور ثبت‌نام می‌کنند. به‌علاوه، هر یک از کاربران می‌توانند با مراجعه به این سرور، لیست تمامی فایل‌های ثبت‌نام شده در سرور را دریافت کنند. مثلاً اگر Peer1 بخواهد File1 را دانلود کند، ابتدا باید به سراغ سرور برود و آدرس‌های هاستی که File1 را در اختیار دارد را دریافت کند. سپس یک اتصال جدید با آن Peer ایجاد کند و فایل را از آن دریافت کند. پس از دریافت فایل، Peer گیرنده نیز خود را به عنوان یکی از میزبانان آن فایل در سرور مرکزی ثبت‌نام می‌کند.

- ارتباط میان یک Peer و سرور مرکزی از نوع UDP می‌باشد

- ارتباط میان Peerها از نوع TCP می‌باشد

- ساختار PDU (Protocol Data Unit):

PDU تبادل شده در سیستم، 101 بایت می‌باشد. بایت اول آن بیان‌گر نوع بسته می‌باشد (در ادامه آن را بررسی خواهیم کرد). صد بایت بعدی نیز برای ارسال صد بایت دیتا استفاده می‌شود.

Type (1byte)	Data (100bytes)
--------------	-----------------

تایپ‌های مختلف PDU به شرح زیر می‌باشد:

- تایپ R: برای ثبت‌نام یک فایل در سرور مرکزی استفاده می‌شود
- تایپ D: برای دانلود یک فایل از یک Peer دیگر استفاده می‌شود
- تایپ S: برای جست‌وجوی یک فایل خاص در سرور مرکزی استفاده می‌شود
- تایپ U: برای حذف یک فایل ثبت‌نام شده از سرور مرکزی استفاده می‌شود
- تایپ F: برای ارسال یک فایل از سوی یک Peer استفاده می‌شود
- تایپ L: برای ارسال آخرین چانک از فایل از سوی یک Peer استفاده می‌شود
- تایپ O: برای دریافت لیست تمام فایل‌های موجود در سرور مرکزی استفاده می‌شود
- تایپ A: برای Acknowledgement کردن بسته‌های دریافتی استفاده می‌شود
- تایپ E: برای PDUهای Error استفاده می‌شود

- ثبت نام فایل:

یک Peer می‌تواند فایل مدنظر خود را با ارسال PDU های تایپ R در سرور مرکزی ثبت نام کند. بخش دیتای PDU شامل موارد زیر می‌باشد:

۱. نام فایل

۲. آدرس هاست (IP + Port)

برای راحتی کار می‌توانید اندازه‌ی تخصیص داده شده به هر یک از موارد فوق را یک مقدار ثابت در نظر بگیرید.

- در صورتی که این فایل قبلاً نیز توسط همین Peer ثبت نام شده باشد، باید یک PDU از تایپ خطا (E) در جواب آن ارسال شود.
- در صورتی که خطایی رخ ندهد، باید یک PDU از تایپ Ack (A) از سوی سرور ارسال شود
- هر Peer به هنگام ارسال بسته‌هایی از نوع R، باید برای فایلی که قصد ثبت نام آن را دارد یک سوکت TCP ایجاد کند و در قسمت Data، مشخصات آن سوکت را قرار بدهد

- دانلود فایل:

برای دانلود یک فایل، ابتدا باید یک PDU از تایپ S به سرور مرکزی ارسال شود. نام فایل در قسمت دیتای این بسته قرار می‌گیرد. در صورتی که چنین فایلی در سرور ثبت نام شده باشد، سرور یک بسته‌ی تایپ S دیگر ارسال می‌کند و در قسمت دیتای آن، آدرس هاستی که میزبان آن فایل می‌باشد را ارسال می‌کند. در صورتی که چنین فایلی در سرور ثبت نام نشده باشد، یک بسته‌ی خطا (تایپ E) در جواب ارسال خواهد شد. (در قسمت دیتای بسته‌های خطا، یک پیغام با معنا ارسال کنید که نشان‌دهنده‌ی نوع خطاست)

در صورتی که Peer درخواست‌دهنده، یک PDU با تایپ S دریافت کند (فایل در سرور ثبت نام شده- باشد)، پس از استخراج آدرس Peer میزبان فایل از بسته‌ی دریافتی، یک اتصال از نوع TCP با Peer دیگر برقرار می‌کند. در صورت موفقیت‌آمیز بودن ایجاد اتصال TCP، یک PDU با تایپ D برای هاست میزبان ارسال می‌شود که در قسمت دیتای آن، نام فایل درخواستی قید شده است. Peer میزبان فایل نیز پس از دریافت این درخواست، فایل را در قالب PDU هایی با تایپ F و با اندازه‌ی 101 بایت ارسال می‌کند. آخرین چانک ارسالی فایل باید تایپ L داشته باشد.

- پس از دریافت فایل، Peer گیرنده باید خودش را به عنوان میزبان جدید فایل، در سرور مرکزی ثبت نام کند.
- یک فایل ممکن است در چند Peer موجود باشد. در این صورت یک مکانیزم ساده‌ی توزیع بار به صورت رندوم ایجاد کنید. یعنی هر بار که یک Peer فایلی را از سرور مرکزی درخواست می‌کند، آدرس یکی از میزبانان آن فایل به صورت تصادفی ارسال شود.
- برای هر Peer یک لیست local از فایل‌هایی که قبلاً ثبت نام کرده‌است نگه‌داری کنید. اگر یک Peer درخواست دانلود فایلی را داد که در local آن موجود است، به او یک پیغام خطا نمایش دهید.

- دریافت لیست تمام فایل‌ها:

با ارسال یک PDU با تایپ O، می‌توان لیستی از تمام فایل‌های ثبت نام شده در سرور مرکزی را از آن دریافت کرد. پاسخ دریافتی از سوی سرور می‌تواند شامل چندین PDU باشد که آخرین PDU تایپ L دارد و مابقی آن‌ها تایپ O دارند.

- حذف یک فایل ثبت نام شده:

برای حذف یک فایل ثبت نام شده، باید یک PDU با تایپ U برای سرور مرکزی ارسال کنیم. در این PDU باید نام فایلی که قصد حذف آن را داریم را ذکر کنیم. در صورتی که حذف موفقیت آمیز باشد، یک بسته با تایپ A در پاسخ ارسال می‌شود. در صورتی که حذف موفقیت آمیز نباشد (مثلاً این فایل اصلاً توسط این Peer ثبت نام نشده باشد)، یک بسته‌ی E در پاسخ ارسال می‌شود.

- ترک سیستم:

به عنوان یک Peer، به هنگام ترک سیستم، ابتدا باید تمام فایل‌های ثبت نام شده را با استفاده از ارسال بسته‌هایی با تایپ U، حذف کنیم و سپس سیستم را ترک کنیم.

- نکات تکمیلی:

- یک Peer باید همواره آماده‌ی پاسخ‌گویی به درخواست Peerهای دیگر باشد. همچنین باید همواره خودش نیز بتواند به عنوان یک Peer از خدمات سیستم استفاده کند و از Peerهای دیگر فایل دریافت کند. برای پیاده‌سازی این دو ویژگی به صورت موازی، می‌توانید از مفاهیم Thread Programming استفاده کنید.
- برای پاسخ‌دهی به درخواست دانلود Peerهای دیگر، می‌توانید کلاً یک سوکت در نظر بگیرید (راهنمایی: از select استفاده کنید). همچنین می‌توانید به ازای هر فایل، یک سوکت مجزا ایجاد کنید. (در این صورت نمره‌ی امتیازی دریافت خواهید کرد.)
- برای پیاده‌سازی PDU می‌توانید از: شی گرای، namedtupleها (در پایتون) و یا structها (در C) استفاده کنید.
- فراموش نشود که ارتباط میان Peerها از نوع TCP و ارتباط بین یک Peer و سرور مرکزی از نوع UDP است.
- برای تست سیستم، می‌توانید از localhost استفاده کنید. (استفاده از Peerهای واقعی، نمره‌ی امتیازی دارد.)
- سیستم را حداقل با چهار Peer تست کنید.
- زبان پیاده‌سازی به انتخاب شما C یا Python می‌باشد.
- سعی کنید در پیاده‌سازی خود، برای هر قابلیت، یک function طراحی کنید.
- پیاده‌سازی خود را به صورت استاندارد و منطبق بر مراحل ذکر شده در درس انجام بدهید.
- یک template برای این سیستم در اختیار شما قرار خواهد گرفت که شما را برای طراحی توابع و ویژگی‌های متفاوت سیستم، راهنمایی می‌کند. می‌توانید پیاده‌سازی را بر اساس آن انجام بدهید و یا طراحی خود را داشته‌باشید.
- یک جلسه‌ی بر-خط برای ارائه‌ی این پروژه در نظر گرفته خواهد شد.
- بارم این مینی‌پروژه، از ۱۲۵۰
- سوالات و ابهامات خود را در گروه درس و یا با ایمیل زیر در میان بگذارید:

Ramtin.Mousavi@Ymail.com

موفق باشید