به نام خداوند بخشنده‌ی مهربان

تمرین برنامه‌نویسی دوم[1]
شبکه‌های کامپیوتری

مدرس: مهدی خرازی **پاییز ۱۳۹۱**

هدف

آشنایی با

- چارچوب کاری پرتو (Partov)
- سرآیندها و سوکت‌های خام
- مفاهیم طراحی شبکه‌های همتا به همتا

۱ مقدمه

امروزه معماری مبتنی بر کارخواه/کارگزار با مشکلاتی از قبیل عدم مقیاس‌پذیری، وجود *تک نقطه‌ی خرابی*[2] و *سربار مدیریتی بالا* روبه‌رو است. به همین دلیل معماران سیستم‌های شبکه‌ای توجه خود را به طراحی سیستم‌های توزیع شده معطوف کرده‌اند. یکی از انواع سیستم‌های توزیع شده، سیستم‌های مبتنی بر محاسبات همتا به همتا[3] می‌باشد که دارای کاربردهای فراوانی در به اشتراک‌گذاری فایل‌ها و محتواهای چندرسانه‌ای هستند. در شبکه‌های همتا به همتا که هر همتا هم کارخواه و هم کارگزار است؛ شبکه‌ی همپوشان[4] با ایجاد یک پیکربندی مجازی بر روی لینک‌های فیزیکی تشکیل می‌شود و با سازوکارهای خاصی قابلیت اتصال و ترک پویای همتاها را فراهم می‌کند. در این تمرین ما به دنبال پیاده‌سازی یک **مخزن داده‌ای توزیع شده‌ی**[5] **مبتنی بر شبکه‌ی همتا به همتا** هستیم که با سانسور کردن اطلاعات مقابله می‌کند. این شبکه، با نگه‌داری فایل‌ها بر روی همتاهای خود و مسیریابی فایل‌ها و درخواست‌ها مبتنی بر تابع درهم‌سازی، می‌تواند ضمن نگه‌داری/بازیابی[6] مجموعه‌ای از فایل‌های سانسور شده، گمنامی کاربران را حفظ کند. همچنین برای حفظ *مقیاس پذیری* در این شبکه سعی شده است مسیریابی‌های صورت گرفته توسط گره‌های میانی از توزیع یکنواخت پیروی کنند که این نیازمند انتخاب تابع درهم‌سازی مناسب و کم و بیش یکنواخت است؛ همچنین برای حذف *تک نقطه‌ی خرابی*، هنگامی که یک همتای جدید وارد شبکه می‌شود وظیفه‌ی نگه‌داری بخشی از اطلاعات شبکه به آن داده می‌شود.

---

[1] با تشکر از با تشکر از بهنام مومنی، مهدی احمدی‌نژاد، کامیار اللهوردی، سجاد فولادی، علی‌محمد ربانی، روزبه کتابی، رامتین رطبی و مهرداد مرادی

[2] Single Point of Failure

[3] Peer-to-Peer Computing

[4] Overlay Networks

[5] Distributed Data Store

[6] Storage and Retrieval

Freenet یک نرم‌افزار متن‌باز[۷] همتا به همتا است که با حذفِ ترس سانسور شدن و محکوم شدن به علت عدم رعایت حقوق انتشار، به شما اجازه‌ی به اشتراک‌گذاری فایل‌های مختلف، انتشار Freesite (وب سایت‌هایی که تنها در داخل این شبکه قابل مشاهده هستند) و چت‌های مخفیانه را می‌دهد. همچنین با توجه به اینکه این شبکه توزیع شده است، ضعف کمتری در مقابل حملات نشان می‌دهد. علاوه بر این، اگر این نرم‌افزار در حالت «شبکه‌ی تاریک[۸]» که در آن کاربران تنها به دوستان خود متصل می‌شوند، استفاده شود، بسیار امن‌تر می‌شود. ما در این تمرین به دنبال پیاده‌سازی نمونه‌ی ساده شده‌ی این نرم‌افزار و پروتکل خاص آن هستیم. در ادامه پس از معرفی محیط، پروتکلی را که باید پیاده‌سازی کنید، شرح می‌دهیم. پیش از مطالعه‌ی بخش‌های بعد، می‌توانید برای آشنایی بیشتر از دو پیوند زیر استفاده کنید:

- http://en.wikipedia.org/wiki/Freenet
- http://freenetproject.org/

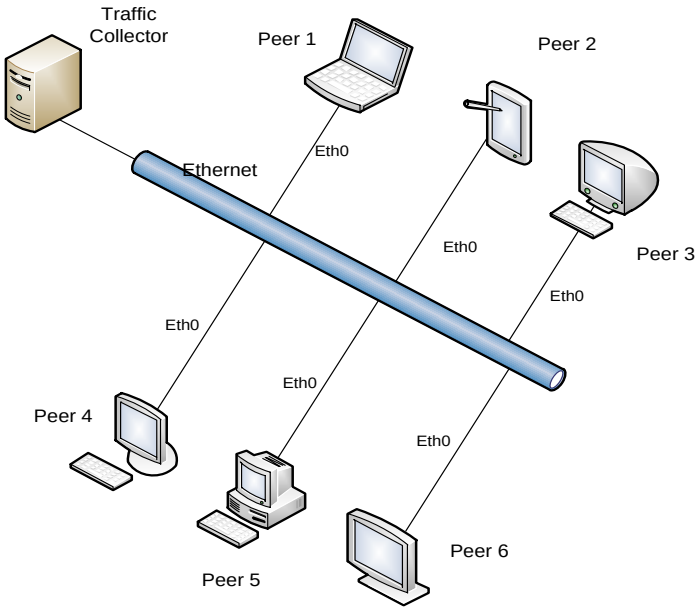
۲ محیط

در این تمرین نیز، مانند تمرین قبل از محیط پرتو[۹] برای اجرای برنامه‌ی شما استفاده خواهد شد. برای این منظور لازم است که برنامه‌های خود را در «چارچوب کاربر[۱۰]» پرتو پیاده سازی کنید. برای اطلاعات بیشتر در این مورد می‌توانید به مستند راهنمای کاربر این محیط مراجعه کنید.

۱.۲ پیکربندی شبکه در محیط «پرتو»

پیکربندی اولیه‌ی شبکه‌ای که در اختیار هر دانشجو قرار می‌گیرد به صورت زیر می‌باشد. شما نباید در برنامه‌ی خود وابستگی خاصی نسبت به پیکربندی داشته باشید و یا فرض خاصی را در مورد پیکربندی در نظر بگیرید. همان‌طور که مشاهده می‌کنید در این پیکربندی هر گره از طریق واسط شماره صفر خود به شبکه‌ی اترنت متصل است. در این شبکه دو نوع گره وجود دارد. دسته‌ی اول گره‌ها، همتاها هستند که با شماره‌های ۱ تا ۶ نام گذاری شده‌اند. دسته‌ی دوم گره‌ها، کارگزارهایی هستند که وظیفه‌ی جمع‌آوری ترافیک همتاها را دارند. در این تمرین تنها یک کارگزار[۱۱] Traffic Collector داریم که دو وظیفه‌ی اصلی دارد:

- همتاها به واسطه‌ی این کارگزار در شبکه ثبت نام می‌کنند، تأییدیه ثبت نام شدن را از این کارگزار دریافت می‌کنند.
- کارگزار تمام ترافیک عبوری از گره‌ها را دریافت می‌کند و به ترتیب زمانی چاپ می‌کند.



«پیکربندی شبکه»

[7] Open Source
[8] Darknet
[9] Portable And Reliable Tool fOr Virtualization (PARTOV)
[10] Client Framework (CF)
[11] هرچند وجود یک کارگزار باعث ایجاد تک نقطه‌ی خرابی می‌شود، ولی با توجه به اینکه خرابی این گره در کارکرد کل شبکه اخلالی ایجاد نمی‌کند، می‌توانیم از مشکل صرف نظر کنیم.

۳ جزییات پروتکل Freenet:

در قسمت قبل در یک شمای کلی پیکربندی شبکه‌ی مورد نظر توضیح داده شد. در این بخش پروتکل این شبکه توضیح داده می‌شود.

۱.۳ سازوکار اتصال همتاها:

هر همتا در Custom Information خود به ترتیب شماره‌ی Portی که باید روی آن منتظر دریافت بسته‌های دیگر همتاها باشد، آدرس IP و شماره‌ی Port کارگزار جمع کننده‌ی ترافیک، آدرس فیزیکی دروازه، تعداد همتاهای پیشنهادی و سپس آدرس IP و Port همتاهای مذکور را دریافت می‌کند. مراحل اتصال همتاها به شرح زیر می‌باشد:

- ***ثبت نام:*** ابتدا هر همتا یک بسته برای درخواست ثبت‌نام به کارگزار ارسال می‌کند و منتظر پاسخ تایید کارگزار می‌ماند. در صورتی که پاسخی دریافت نکرد این کار را بعد از ۵ ثانیه دوباره تکرار می‌کند تا زمانی که کارگزار پاسخ تایید را برایش ارسال کند.

- ***پیدا کردن همسایه:*** هر همتا به ترتیب به هر یک از همتاهای احتمالی موجود در Custom Information بسته‌ای را مبنی بر همسایه شدن ارسال می‌کند. در صورتی که همتای مقصد پاسخ تایید بدهد، آن را به عنوان همسایه در نظر می‌گیرد. اما اگر همتای مقصد پاسخ ندهد، بعد از ۳ ثانیه همتای مبدأ درخواست همسایه شدن را دوباره ارسال می‌کند. اگر برای بار دوم نیز پاسخی دریافت نشد، دیگر درخواست ارسال نمی‌شود و این یعنی اینکه همتای مورد نظر علاقه ای به همسایه شدن با ما ندارد.

Freenet Peer Custom Information
Port Number to Listen
Traffic Collector IP Address/ Port Number
Gateway Mac Address
Number of Possible Peers
IP Address/Port Number of Possible Peers

در زیر یک نمونه از اطلاعات آورده شده است:

مثال:

Freenet Peer Custom Information
53021 213.233.176.17 2347 01:23:45:67:89:ab 3 213.233.167.19 1234 213.233.168.18 2234 213.224.167.19 1890

همچنین Custom Information کارگزار در زیر آورده شده است که تنها شامل آدرس فیزیکی دروازه می‌باشد.

Traffic Collector Custom Information
Gateway Mac Address

مثال:

Traffic Collector Custom Information
01:23:45:67:89:ab

۲.۳ سازوکار بازیابی یک فایل:

برای بازیابی یک فایل، کاربرِ همتا باید یک درخواست[۱۲] به همتای خود با مشخص کردن مقدار hops-to-live و نام فایل ارسال کند. به منظور پیدا کردن فایل مورد نظر، این پیام درخواست در شبکه‌ی پوشش، توسط همتاهای مختلف مسیریابی می‌شود، تا اینکه یا فایل مورد نظر باز گردانده شود و یا اینکه مشخص شود این فایل در شبکه وجود ندارد. در نتیجه، سازوکار بازیابی یک فایل شامل دو مولفه‌ی ***الگوریتم مسیریابی*** و ***الگوریتم بازیابی*** است که در ادامه توضیح داده شده‌اند:

۱.۲.۳ الگوریتم مسیریابی:

فرض کنید یک همتای خاص می‌خواهد یک پیام بازیابی را برای یکی از همسایگانش ارسال کند. الگوریتم مسیریابی مشخص‌کننده‌ی این است که این پیام باید به کدام همسایه ارسال گردد. باید توجه کرد که انتخاب صحیح همسایه می‌تواند میزان مصرف و الگوی مصرف منابع شبکه را تغییر دهد. در نتیجه مسیریابی صحیح یکی از مهم‌ترین بخش‌های پروتکل Freenet است.

هر همتا در شبکه یک جدول مسیریابی دارد که در ابتدا خالی است و به مرور زمان در آن مدخل‌هایی ایجاد می‌کند. این مدخل‌ها شامل دو مولفه می‌باشند. مولفه‌ی اول آدرس IP همسایگان و مولفه‌ی دوم شناسه‌ی فایل‌های دریافتی از همسایه مذکور است که شناسه‌ی فایل‌ها از درهم‌سازی نام فایل‌ها به دست می‌آید.

مثال:

جدول مسیریابی زیر مربوط به یک همتا است که چهار همسایه دارد که از همسایه‌های خود تعدادی فایل را دریافت کرده است.

جدول مسیریابی	
شناسه‌ی فایل‌های دریافتی	**آدرس IP**
۴- ۲۴- ۱۳- ۱۲	۲۱۳.۲۳۳.۱۶۸.۱
۳۴-۴۴	۲۱۳.۲۳۳.۱۶۹.۲

هنگامی که یک همتای دلخواه می‌خواهد یک درخواست بازیابی را ارسال کند، مراحل زیر طی می‌شود (**البته دقت شود درخواست بازیابی به همتایی که درخواست را از او دریافت کرده‌ایم ارسال نمی‌شود**):

1. ***اگر جدول مسیریابی خالی باشد:***
   - در این صورت درخواست مذکور به همسایه‌ای ارسال می‌شود که مقدار زیر برای آن نسبت به دیگر همسایه‌ها کمینه باشد.

$$|hash(filename) - hash(IP\ of\ Neighbor)|$$

2. ***اگر جدول مسیریابی خالی نباشد:***
   - در این صورت درخواست مذکور به همسایه‌ای ارسال می‌شود که قبلاً از آن فایلی دریافت کرده باشیم که مقدار زیر برای آن فایل نسبت به دیگر همسایه‌ها کمینه باشد.

$$|hash(filename) - id\ of\ file|$$

مثال:

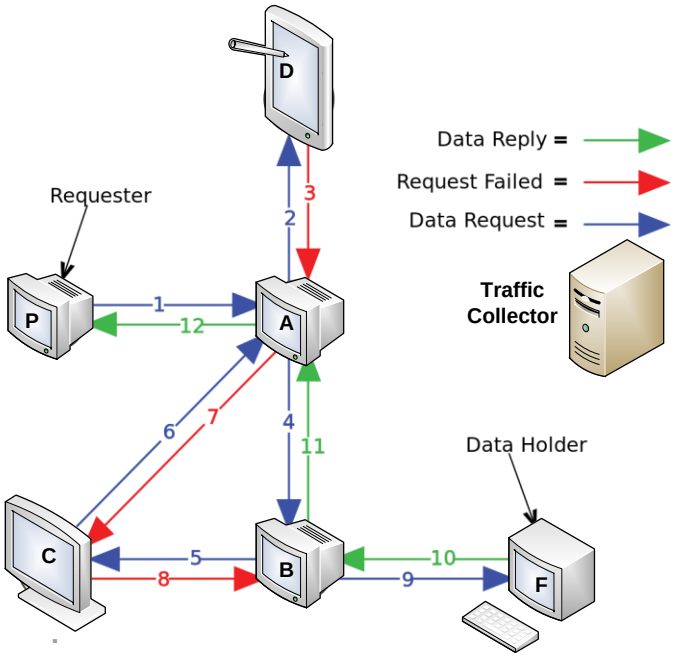
فرض کنید که جدول مسیریابی بالا را داریم و شناسه‌ی فایل درخواستی برابر با ۲۸ است. در این صورت با توجه به اینکه فاصله ۲۸ با ۲۴ از همه کمتر است درخواست مورد نظر به ۲۱۳.۲۳۳.۱۶۸.۱ ارسال می‌گردد.

---

[۱۲] این درخواست از طریق ترمینال و با قالب شرح داده شده در بخش ۱.۴ به همتا ارسال می‌شود و همتا در صورت نیاز با ارسال بسته‌های مناسب به درخواست کاربر رسیدگی می‌کند. طراحی قالب بسته‌های ارسالی به عهده‌ی خودتان است.

۲.۲.۳ الگوریتم بازیابی:

در این مرحله صرف نظر از اینکه کاربر کدام همتا درخواست بازیابی را ایجاد کرده است، فرض می‌کنیم که یک همتای دلخواه، چنین درخواستی را دریافت کرده است.



این همتا ابتدا مخزن داده‌ای خود را برای فایل مذکور جست و جو می‌کند، در ادامه:

۱. ***اگر فایل مورد نظر را یافت:***

- فایل مورد نظر را به همتایی که از آن درخواست را دریافت کرده است بر می‌گرداند. در صورتی که خودش ایجاد کننده درخواست باشد، رشته‌ی Have it! را در کنسول چاپ می‌کند و در صورتی که همتای دیگری درخواست را برای اولین بار ایجاد کرده باشد، آن همتا پس از دریافت فایل، رشته‌ی Received From Others! را چاپ می‌کند.

۲. ***اگر فایل مورد نظر یافت نشد:***

- بر اساس الگوریتم مسیریابی بیان شده، درخواست بازیابی را ضمن کم کردن hops-to-live به اندازه یک واحد، به یکی از همسایگانش ارسال می‌کند. هرکدام از همتاها هنگامی که درخواست بازیابی را دریافت کنند، مراحل ۱ و ۲ را تکرار می‌کنند. این عمل آن قدر اتفاق می‌افتد تا اینکه:

- مقدار hops-to-live صفر شود: در این صورت هر همتایی که این مقدار را مشاهده کند به همتایی که درخواست را از آن دریافت کرده است، پیغامی مبنی بر عدم وجود فایل ارسال می‌کند. این عمل آن قدر اتفاق می‌افتد تا ایجاد کننده‌ی درخواست، پیغام Not Found! را دریافت کند.
- یک همتای خاص فایل مورد نظر را پیدا کند: در این صورت فایل مذکور را به همتایی که درخواست را از آن دریافت کرده است ارسال می‌کند. هر همتایی که در مسیر، فایل مذکور را مشاهده می‌کند، علاوه بر این که فایل را به همتایی که از آن درخواست کرده ارسال می‌کند، جدول مسیریابی خود را به روز رسانی می‌کند و یک کپی از فایل را نیز نگه‌داری می‌کند.
- یک همتای خاص اطمینان حاصل کند که نه خودش و نه همسایگانش فایل درخواستی را ندارند: در این صورت پیغامی مبنی بر عدم وجود فایل به همتایی که از آن درخواست کرده ارسال می‌کند.

با این نوع سازوکار بازیابی فایل، تعدادی گره می‌توانند ادعا کنند که منبع پخش فایل بوده‌اند، **در نتیجه همتای منبعِ فایل همواره گمنام می‌ماند.**

۳.۳ اضافه کردن یک فایل[۱۳]:

برای این که از بارگذاری دوباره یک فایل در شبکه جلوگیری شود، هر همتایی که می‌خواهد یک فایل را بارگذاری کند، در ابتدا اطمینان حاصل می‌کند که فایل مذکور در شبکه وجود ندارد و سپس اقدام به بارگذاری می‌کند. به صورت دقیق‌تر مراحل زیر طی می‌شود:

- تمامی مراحلی که برای بازیابی فایل گفته شد انجام می‌شود، با این تفاوت که فایل مذکور به همتای ایجاد کننده‌ی درخواست بازیابی، انتقال پیدا نمی‌کند و رشته ای هم بر روی صفحه چاپ نمی‌شود.
  - ✓ در صورتی که فایل مذکور در مرحله‌ی قبل پیدا شد: همتایی که می‌خواهد فایل را در شبکه اضافه کند، پیغامی مبنی بر وجود فایل چاپ می‌کند.
  - ✓ در صورتی که فایل مذکور پیدا نشد: فایل مذکور را در لیست فایل‌های خود قرار می‌دهد. در نتیجه اگر همتای دیگری بخواهد می‌تواند این فایل را بازیابی کند.

۴.۳ سازوکار کنترل ترافیک:

همان طور که در بخش ۱.۲ توضیح داده شد، برای اینکه بتوانیم صحت کارکرد شبکه‌ی Freenet را ارزیابی کنیم، کارگزاری را در این شبکه به نام Traffic Collector قرار داده‌ایم که به واسطه‌ی آن می‌توانیم تمامی بسته‌هایی را که در شبکه رد و بدل می‌شوند با ترتیب زمانی ثبت کنیم. به این منظور هنگامی که یک گره هر یک از عملیات گفته شده در بخش ۵ را انجام داد، باید بسته‌ای حاوی اطلاعات فرستنده، گیرنده و نوع عملیات را برای این کارگزار ارسال کند. در نتیجه کارگزار خواهد توانست گزارش دقیقی از رخدادهای درون شبکه Freenet را نمایش دهد.

[۱۳] در تمامی قسمت‌های مرتبط با بارگذاری فایل، فرض بر این است که کل فایل مورد نظر قابل ارسال توسط یک بسته است، گرچه فرضی مبنی بر متنی بودن فایل‌ها نیست.

۴ برنامه‌ی مربوط به گره‌ها:

حال که با سازوکار این شبکه آشنا شدیم، از شما انتظار داریم که شبکه توصیف شده را در محیط بیان شده پیاده‌سازی کنید، به نحوی که کاربرانِ همتاها و کاربرِ کارگزار بتوانند دستورات زیر را اجرا کنند.

۱.۴ دستورات مربوط به همتاها:

برنامه مربوط به هر گره باید بتواند دستورات زیر را پشتیبانی کند.

۱. دستور **Join**:

با وارد کردن این دستور در ترمینال مربوط به یک همتا، آن همتا باید وارد شبکه شود. برای این منظور لازم است شما طبق اطلاعات موجود در بخش **سازوکار اتصال** به شبکه متصل شوید. هنگامی که اتصال صحیح صورت گرفت شما باید رشته زیر را چاپ کنید:

Input	Join
Output	Joined to Freenet!

۲. دستور **Insert**:

این دستور برای ذخیره کردن یک فایل در شبکه می‌باشد. در ادامه‌ی این دستور، نام فایل مورد نظر و hops-to-live آورده شده است. برای مثال می‌توان روی یکی از گره‌ها دستور زیر را اجرا کرد:

Input	Insert MehrdadInfinity.pdf 3

۳. دستور **Retrieve**:

از این دستور برای بازیابی یک فایل استفاده می‌شود. بعد از وارد کردن این دستور، نام فایل مورد نظر و سپس hops-to-live آورده می‌شود. به عنوان مثال در حالت اول مشاهده می‌کنید که فایل وجود ندارد یا اینکه hops-to-live تمام شده است. حالت دوم برای زمانی هست که فایل از دیگران دریافت گردد. خروجی‌های ممکن در زیر قابل مشاهده است:

Input	Retrieve LostPacket.mkv 4
Outputs	Not Found!
	Received From Others!
	Have it!

۴. دستور **PrintNeighbors**:

با وارد کردن این دستور باید اطلاعات مربوط به اطلاعات همسایه‌ها در خروجی استاندارد چاپ شوند. به مثال زیر توجه کنید:

Input	PrintNeighbors
Output	192.168.6.2 192.168.6.5

اگر یک گره همسایه‌ای نداشت، عبارت « I am alone!» برایش چاپ می‌شود.

۵ برنامه‌ی مربوط به گره‌ی جمع‌آوری کننده‌ی ترافیک:

این برنامه باید بتواند دستورات زیر را اجرا کند.

۱. دستور **PrintTraffic**:

با وارد کردن این دستور گره‌ی جمع آوری کننده‌ی ترافیک باید تمامی ترافیک عبوری را با توجه به زمان وقوع نمایش دهد. لیست وقایعی که باید ثبت شوند در زیر آمده است:

Event Name	Description
JoinRequest	Request for join is being sent
JoinAccepted	Accepting join request
FileRequest	Request for a file is being sent
DataReply	Sending file contents
FailedFileRequest	Sending a failed file request message

نمونه‌ای از این گزارش در زیر آمده است (تنها بخشی از اطلاعات نشان داده شده است):

Input	PrintTraffic
Output	From: 192.168.11.10 To: 192.168.12.10 Type: JoinRequest From: 192.168.12.10 To: 192.168.11.10 Type: JoinAccepted From: 192.168.11.10 To: 192.168.13.10 Type: JoinRequest From: 192.168.13.10 To: 192.168.11.10 Type: JoinAccepted From: 192.168.11.10 To: 192.168.14.10 Type: JoinRequest From: 192.168.11.10 To: 192.168.14.10 Type: JoinRequest From: 192.168.11.10 To: 192.168.12.10 Type: FileRequest From: 192.168.12.10 To: 192.168.13.10 Type: FileRequest From: 192.168.13.10 To: 192.168.12.10 Type: FailedFileRequest From: 192.168.12.10 To: 192.168.11.10 Type: FailedFileRequest From: 192.168.13.10 To: 192.168.12.10 Type: FileRequest From: 192.168.12.10 To: 192.168.11.10 Type: FileRequest From: 192.168.11.10 To: 192.168.12.10 Type: DataReply From: 192.168.12.10 To: 192.168.13.10 Type: DataReply

۱- دستور **PrintPeers**:

با این دستور آدرس IP تمام همتاهایی که در کارگزار ثبت نام شده‌اند، چاپ می‌شود. اگر همتایی ثبت نام نکرده بود No Peers! چاپ می‌شود. به مثال زیر توجه کنید:

Input	PrintPeers
Output	192.168.6.2 192.168.6.5

نکات ضروری

- در صورتی که هر مشکلی یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، آن را به گروه پستی درس ارسال کنید.
- از فرستادن تمرین به گروه پستی خودداری کنید.
- فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فرد دیگری به نام خود، تقلب محسوب می‌شود.
- پس از اتمام کارتان، لازم است که پوشه‌ی user را به همراه Makefile فشرده کرده و بر روی سیستم خودکار داوری[14] بارگذاری کنید. از ارسال فایل‌های باینری خودداری کرده و حتماً پیش از ارسال make clean را اجرا نمایید.

موفق باشید.

[14] http://partov.sharif.edu/