## **بسمه تعالی**



**گزارش تمرین 4**

**شبکه‌های کامپیوتری**

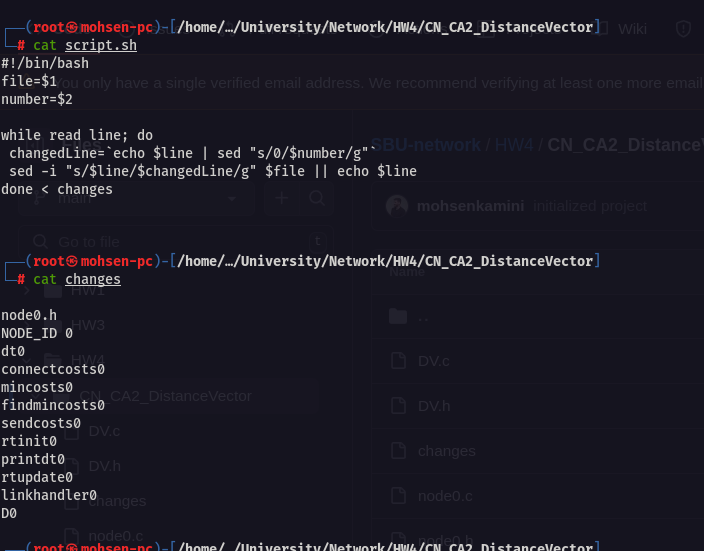
محسن کربلائی امینی، 98242128

تیر 1403

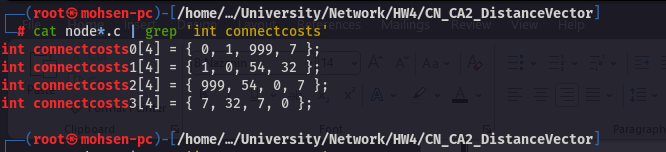
**تنظیم مقادیر اولیه و آماده‌سازی کد‌های مورد نیاز:**

برای آماده‌سازی کد در ابتدا نیاز هست تا کد‌های مربوط به گره‌های جدید ساخته شود. کد اصلی node0.h و node0.c را برای هر کدام کپی کرده و سپس با بهره گیری از فایل‌های changes و script.sh هر فایل آماده کامپایل می‌کنیم.  


محتویات این دو فایل کمکی:

****

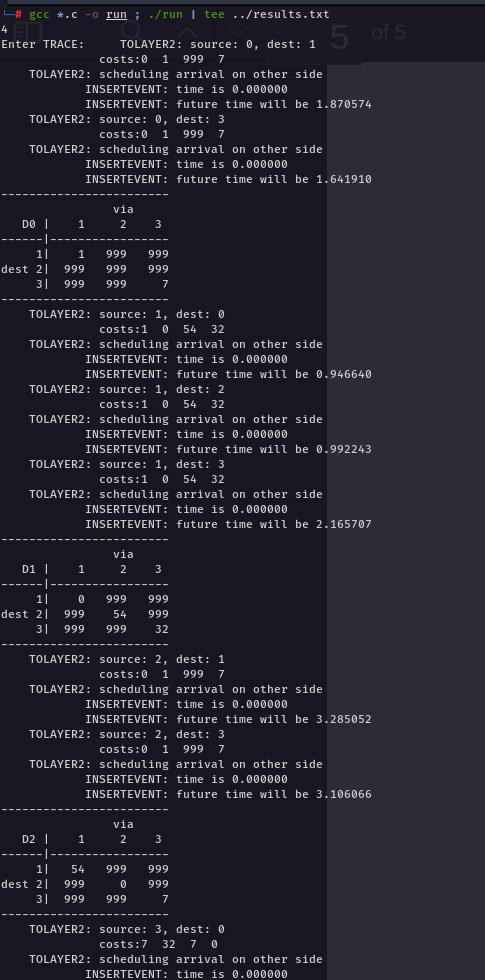
و در آخر تنظیم مقادیر اولیه مربوط به هزینه به گره‌های متصل:



**توضیح ارتباطات:**

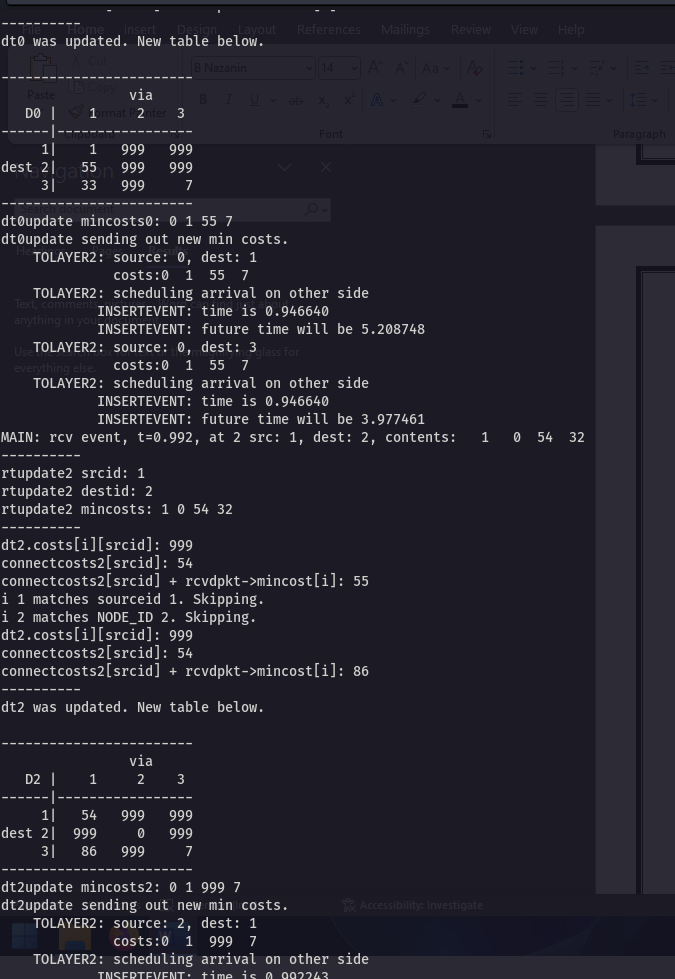
**مرحله اول: شناخت خود**

در این مرحله جداول اولیه هر گره تولید می‌شود. مثلا برای گره شماره 0 مشخص می‌شود که به هر گره دیگر از چه طریقی می‌تواند عبور کند و اگر یک گره‌ دیگر را برای رسیدن به آن گره واسطه قرار دهد، چه هزینه‌ای را به همراه دارد. برای مثال اگر گره 0 در این مرحله بخواهد با کمک گرفتن گره 1 به مقصد 3 برسد، این کار امکان‌پذیر نیست چرا که هنوز از مسیر‌های گره شماره 1 خبر ندارد. بنابراین فقط از مسیر‌های مستقیمی که خودش لینک مستقیم دارد می‌تواند استفاده کند. برای باقی هم به همین شکل:



**مراحل میانی: شناخت دیگران و ارتباطاتشان**

در این مراحل به مرور زمان گره‌ها در هر مرحله اطلاعات مسیر‌یابی را از دیگران گرفته و تلاش می‌کنند تا جداول خود را بهینه‌ کنند. برای مثال گره 0 که در مرحله اول نمی‌توانست از طریق گره 1 به گره 3 برسد بعد از گذشت مرحله شناخت، با ارتباط گرفتن با گره 1 متوجه می‌شود که این گره لینکی با هزینه 32 با گره 3 دارد. بنابراین 32+1 که هزینه رسیدن از 0 به 1 است، مقدار هزینه 33 را برای این مسیر تعیین می‌کند:



**مرحله اشباع:**

در این مرحله تمام بهینه‌سازی‌ها در جداول مسافت هر گره اتفاق افتاده. البته در دنیای واقعی این شرایط همیشه پایدار نخواهد بود و به محض آپدیت هزینه یا اضافه شدن مسیری به هر یک از این گره‌ها، دوباره ارتباط‌ها برای بهینه‌سازی جداول مسافت از سر میگیرد.  
اما در این سناریو در نهایت به جداول زیر خواهیم رسید:



**توضیح توابع کد:**

void findmincosts0():

این تابع وظیفه به‌روزرسانی مقادیر موجود در آرایه mincosts0 را بر عهده دارد.

ابتدا این تابع یک حلقه برای هر یک از چهار گره انجام می‌دهد (به جز گره خودش).

سپس برای هر یک از چهار گره، یک حلقه دیگر برای چهار گره انجام می‌دهد (به جز گره خودش).

در این حلقه‌ها، اگر هزینه مسیر موجود در جدول مسافت (dt0.costs[i][j]) کمتر از مقدار فعلی mincosts0[i] باشد، مقدار mincosts0[i] به هزینه جدید بروزرسانی می‌شود.

void sendcosts0():

این تابع مسئول ارسال هزینه‌های کمینه به همسایگان مستقیم گره است.

ابتدا یک ساختار rtpkt به نام cost\_pkt ایجاد می‌کند که حاوی مقادیر sourceid، destid و mincost است. سپس یک حلقه برای هر یک از چهار گره (به جز خودش و گره‌هایی که به طور مستقیم به آن‌ها متصل نیستند) انجام می‌دهد. در این حلقه‌ها، مقدار destid در ساختار cost\_pkt تنظیم شده و توابع tolayer2() برای ارسال بسته فراخوانی می‌شود.

void rtinit0():

این تابع مسئول فراهم‌سازی اولیه جدول مسافت است.

ابتدا یک حلقه برای هر یک از چهار گره انجام می‌دهد. سپس برای هر گره، یک حلقه دیگر برای چهار گره اجرا می‌شود. اگر گره مبدأ و مقصد یکی باشند، هزینه مربوطه از connectcosts0 گرفته می‌شود. در غیر این صورت، هزینه به مقدار 999 تنظیم می‌شود.

پس از تنظیم جدول مسافت، تابع sendcosts0() فراخوانی می‌شود تا هزینه‌های کمینه به همسایگان ارسال شود.

در نهایت، تابع printdt0() فراخوانی می‌شود تا جدول مسافت چاپ شود.

rtupdate0(struct rtpkt \*rcvdpkt):

این تابع زمانی فراخوانی می‌شود که یک بسته مسیریابی دریافت شود.

ابتدا اطلاعات بسته مسیریابی دریافتی (مانند شناسه منبع و مقصد و هزینه‌های مسیر) را چاپ می‌کند.

سپس بروزرسانی جدول مسیریابی را انجام می‌دهد. اگر هزینه مسیر جدید کم‌تر از هزینه مسیر قبلی باشد، جدول مسیریابی بروزرسانی می‌شود.

در صورت بروزرسانی جدول مسیریابی، هزینه‌های کمینه مسیر به‌روز شده و به همسایه‌ها ارسال می‌شود.

printdt0():

این تابع جدول مسیریابی dt0 را چاپ می‌کند.

جدول شامل هزینه مسیرها از هر گره به بقیه گره‌ها است.

linkhandler0(int linkid, int newcost):

این تابع زمانی فراخوانی می‌شود که هزینه یک پیوند تغییر کند. ابتدا هزینه‌های جدید را چاپ می‌کند و

سپس جدول مسیریابی را به این شکل بروزرسانی می‌کند:

* هزینه مستقیم پیوند را بروز می‌کند.
* هزینه پیوندهای غیرمستقیم از طریق آن پیوند را به صفر تنظیم می‌کند.
* هزینه‌های کمینه مسیر را بروز و محاسبه می‌کند.

در نهایت جدول مسیریابی بروزرسانی شده را به همسایه‌ها ارسال می‌کند.