CS360 -- Lab 2 -- Buffering

- Jian Huang
- CS360
- Url: http://www.cs.utk.edu/~huangj/cs360/360/labs/lab2/lab.html

---

Bu laboratuar read() fread() ve buffering ile ilgili bize pratik verir.

İlk olarak, ne olursa olsun kod yazacağınız dizinde, aşağıdakileri takip edin:

```
UNIX> ln -s ~huangj/cs360/labs/lab2/hosts
UNIX> ln -s ~huangj/cs360/labs/lab2/converted
```

Bu ana dosyalarla bağlantıları yapacak, ve laboratuar dizinimde hangisi olduğunu döndürecek. Biz bunun yerine disk alanına kaydetmek için bir kopyasıyla bağlantı istiyoruz. Bu atama ile işimiz bittiğinde, bu bağlantılar ve bu dosyaların tüm kopyalarını siliyoruz. Çünkü bu dosyalar sırasıyla 335K ve 488K'dır. *Bu çoğu öğrencinin, bu dosyaları kopyalamasını engeller ve disk boşluğunu öldürerek öğrencilere ayırır.(*Bu cümle eksik olmuş olabilir.)

Şimdi, ana dosyalara bakarak:

```
UNIX> head hosts
128.169.202.200 cocker.vet.utk.edu      cocker
128.169.201.208 berkelium.chem.utk.edu  berkelium
128.169.201.152 cherokee.gg.utk.edu     cherokee
192.31.99.193   slipgate1.utk.edu       slipgate1
192.249.11.79   csmacw14.cs.utk.edu     csmacw14
128.169.94.12   cygnusx1.cs.utk.edu     cygnusx1
128.169.93.24   pepsi.cs.utk.edu        pepsi
128.169.92.48   wham-o.cs.utk.edu       wham-o
127.0.0.1       localhost       loghost
128.169.202.248 mccall.oac.utk.edu      mccall
UNIX>
```

Host dosyasının her satırı bir makineye UT, ORNL veya Princeton' a atıfta bulunur. Her bir makinenin satırdaki ilk sözcüğü IP adresidir. Bu adres, 0 ve 255'in arasında dört sayıdan oluşur ve makinelerin internet üzerinde nasıl erişilebilir olduğunu gösterir. Sonraki tüm isimler bu makine tarafından bilinen isimlerdir. Çoğu makinede dikkat edeceğimiz iki kısım vardır – yerel ad (noktalar olmadan, örneğin."cocker"), ve mutlak ad (noktalarla beraber, örneğin."cocker.vet.utk.edu").

Bazı makinalar sadece yerel adlara sahiptirler("localhost gibi"), ve bazı makinalar birden fazla yerel ad a sahiptirler(128.112.128.1 gibi, hangi vasıtayla çalıştığı "princeton.edu" ,"princeton", ve "newserver"). bir makinenin mutlak bir ad a sahip olduğunda, bu mutlak (örneğin, `` cocker'''' cocker.vet.utk.edu ``) şeklinde olabilir.
it also is known by a local name which is the prefix of the absolute name (e.g. ``cocker'' is the prefix of ``cocker.vet.utk.edu'').

Şimdi , Dönüştürülen dosya host gibi aynı bilgileri içerir. Sadece içinde bir terser form vardır. **Dönüştürülen** sadece belirli bir biçimde bağlı kalarak bayt akışıdır.Host gibi, her makine için art arda girdileri içerir; ancak, **dönüştürüleni head** veya **vi** ile okuyamazsınız. Bunun yerine bunu okumak için bir program yazmak gerekir. Her makinenin aşağıdaki giriş formuna sahiptir:

- İlk dört baytta IP adresi vardır . Diğer bir deyişle, **dönüştürülen** ilk dört byte, 200 128, 169, 202, vardır --- bu'' cocker `` bilgisayarın IP adresidir.
- Sonraki dört bayt tamsayı olarak yorumlanır. Bu, tam sayı makinesi bilinen tarafından isimlerin numarası içerir.
- Sonraki byte makine adlarını içerir, boş sonlandır.
- Bir sonraki madde son null karakterinden hemen sonra başlar.

Bir girişi mutlak bir makine adı içeriyorsa, yerden kazanmak için, o mutlak adının öneki karşılık yerel adı atlanmış kabul edilir. Böylece, ilk 58 byte makineler için bilgileri içeren dönüştürülen `` cocker.vet.utk.edu'' ve `` berkelium.chem.utk.edu'':

x	Bayt *x*	Bayt *x+1*	Bayt *x+2*	Bayt *x+3*	Bayt *x+4*	Bayt *x+5*	Bayt *x+6*	Bayt *x+7*
1	128	169	202	200	0	0	0	1
9	99 ('**c**')	111 ('**o**')	99 ('**c**')	107 ('**k**')	101 ('**e**')	114 ('**r**')	46 ('**.**')	118 ('**v**')
17	101 ('**e**')	116 ('**t**')	46 ('**.**')	117 ('**u**')	116 ('**t**')	107 ('**k**')	46 ('**.**')	101 ('**e**')
25	100 ('**d**')	117 ('**u**')	0 ('**\0**')	128	169	201	208	0
33	0	0	1	98 ('**b**')	101 ('**e**')	114 ('**r**')	107 ('**k**')	101 ('**e**')
41	108 ('**l**')	105 ('**i**')	117 ('**u**')	109 ('**m**')	46 ('**.**')	99 ('**c**')	104 ('**h**')	101 ('**e**')
49	109 ('**m**')	46 ('**.**')	117 ('**u**')	116 ('**t**')	107 ('**k**')	46 ('**.**')	101 ('**e**')	100 ('**d**')
57	117 ('**u**')	0 ('**\0**')						

Bölüm 1

Program l2 p1 yazın. Bu program, makine isimleri anahtarlı bir kırmızı-siyah ağaç dönüştürülmüş dosyayı ve ana bilgisayar bilgilerini okur.Dosyasını da okuduktan sonra, kullanıcı bir makine adı girmenizi ister ve sonra IP adresi ve makine için tüm isimleri yazdırır.Kırmızı-siyah ağaç yerel ve mutlak adlarını içermelidir. Ayrıca bu isimlerin dönüştürülen dosyasında (olmasalar bile, mutlak bir isim önekleri yerel adlarını içermelidir yani `` cocker'' ve `` berkelyum'' değillerse bile, kırmızı-siyah ağaç olmalıdır açıkça dönüştürülür dosyasında). Bu dönüştürülen dosyada yinelenen yerel isimler

olduğunu varsayabiliriz.Dönüştürülen dosyada okumak için tamponlu (standart) I / O rutinleri kullanın. Diğer bir deyişle, fgetc (), fscanf (), vb fread () kullanın --- sizin için hangisi uygunsa. ~ cs360/lab2/l2p1 sizinki gerektiği gibi çalıştırılabilir bir dosyadır. Aşağıdaki örnek, l2p1 gelen girdi ve çıktısı:

```
UNIX> ~cs360/lab2/l2p1
Hosts all read in

Enter host name: hydra3e
128.169.94.137: hydra3e.cs.utk.edu hydra3e

Enter host name: sun4server
128.112.130.225: commonserver decmipsserver decmserver
diskfarm.princeton.edu diskfarm homeserver irisserver mailserver
oedfileserver rtpcserver sun3server sun4server

Enter host name: duncan
128.169.94.83: duncan.cs.utk.edu duncan

Enter host name: xxx
no key xxx

Enter host name: < CNTL-D >
UNIX>
```

Eğer l2p1 çalıştırmayı denerseniz de şöyle der:

12p1: değiştirilmiş: No such file or directory

Sonra **~huangj/cs360/labs/lab2/converted** linki yapmanıza gerek yok.

```
UNIX> ln -s ~huangj/cs360/labs/lab2/converted
```

yapın ve tekrar deneyin. Farkedeceksiniz ki **converted** dosyadaki l2p1'i okumak birkaç saniye alır. Kullanım süresini **time** ile ayarlayabilirsiniz ve boş dosya olarak **/dev/null**'a girdi yapabilrsiniz.

```
UNIX> time ~cs360/lab2/l2p1 < /dev/null
Hosts all read in

Enter host name: 3.4u 0.6s 0:04 99% 0+1824k 1+0io 0pf+0w
```

Burada programın süresinin kullanıcı süresi olarak 3.4 saniye, sistem süresi olarak 0.6 saniye ve duvar saati süresi olarak 0:04 (4 saniye) alacağı söylenmektedir. Programınız karşılaştırılabilir bir süreye sahip olmalıdır(3-5 saniye arası, buradaki program hızlı olacaktır çünkü **gcc**'nin –**O** bayrağı ile derlendi. Bu bayrak, kodu mümkün olduğunca hızlı bir şekilde kodlamaya çalışır. İsterseniz siz de deneyebilirsiniz.)

Bölüm 2

l2p2 programını yazın. Bu program converted dosyasındaki sistem çağrıları **open(),close()** ve **read()**kullanman dışında kesinlikle **l2p1** ile aynıdır.(Converted dosya dışındaki standart input ve standart output dosyalarını okumak /yazmak için buffered I/O kullanabilirsiniz). Verimlilik konusunda

endişelenmeyin, sadece programı çalıştırın. Doğru bir yapı oluşturduysanız, **~cs360/lab2/l2p2** şeklindeki sizinki ile aynı çalışan bir uygulanabilir dosya için birkaç modifikasyon sürecektir.

Not: Bu l2p1'den oldukça yavaştır.

```
UNIX> time ~cs360/lab2/l2p2 < /dev/null
Hosts all read in

Enter host name: 4.7u 21.9s 0:26 99% 0+1796k 5+0io 5pf+0w
UNIX>
```

30 – 40 saniye aralığında oldukça iyi çalışma süreleri elde edebilirsiniz.

Bölüm 3

Program l2p3'ü yazın. İşiniz l2p2'yi alıp l2p1 olarak çalıştırma yapmaktır. Open(), close() ve read()'i kendi tamponlarına eklemesi gerekir. Bu, ben sadece bir kez read()'i çağıracağım demektir. Aynı anda dönüştürülmüş 350000 byte'ı tampondan okuyabilirsiniz. Çalışma buradan yürütülebilir **~cs360/lab2/l2p3:**

```
UNIX> time ~cs360/lab2/l2p3 < /dev/null
Hosts all read in

Enter host name: 3.0u 0.7s 0:04 76% 0+2064k 41+0io 41pf+0w
UNIX>
```

Advice

Her zamanki gibi, bu laboratuara başlarken küçük şeylerle başlamalı ve bir yol oluşturulmalı. Hatta başarıyla dönüştürülen dosyayı okuyun ve içeriğini yorumlayın. Sonra rb-ağacı için endişe başlar. Sizin rb-ağacı makine adı anahtarlı ve aşağıdaki gibi bir yapı işaret eden bir val.v alana sahip olmalıdır:

```
struct ip {
  unsigned char address[4];
  Dllist names;
};
```

Eğer isimlerin alfabetik çıktısına sahip olmak istiyorsanız, bir rb-ağacı yerine dllist(çift yönlü liste) kullanın. İlk rb-ağaç kodu yazmaya başladığınızda, yapı uğraşmıyor. Bunun yerine, sadece ip adresi val.v alanına sahip olmak gerek. Sonra rb-ağaç için bir yapı ekleyin. Rb-ağaç kodunuz üç parça için de aynı olmalıdır.

Eğer bir l2p3 yol hatası alırsanız

İlk l2p3 çalıştığınızda şansınız iyiyse, yol hatası alırsınız. Çünkü bu bir tamsayı olarak tampon bir giriş için isim sayısına erişmeye çalışacaktır, ama uyumlu olmayacaktır. Örneğin, dönüştürülmüş bir

tampon s olduğunu ve ikinci giriş için isim numaralarının (s+13)'den az olduğunu varsayalım. Gibi bir şey deneyin:

```
char *s;
  int *i, count;

  i = (int *) (s+13);
  count = *i
```

O zaman bir yol hatası alırsınız. Bunun yerine, memcpy(&count, s+13, sizeof(int)) kullanın. Bu sizin için kafa karıştırıcı ise man sayfasını okuyun.