

# CS360 Lecture notes -- Prsize: recursive directory traversal

- [Jim Plank](#)
- Directory: `/blugreen/homes/plank/cs360/notes/Prsize`
- Lecture notes: <http://www.cs.utk.edu/~plank/plank/classes/cs360/360/notes/Prsize/lecture.html>

---

Bu notlar, bir **prsize** command'ını yazmak üzerinedir. **prsize**, mevcut 'directory'den erişilebilir olan tüm dosyalardan (soft linkler dışında) alınan byte sayısının geri dönüşünü ifade eder.

**opendir/readdir/closedir**, **stat**, recursion kullanmayı canlandırdığı ve uzantı isimlerini inşa etmesi ve hard linkleri bulması açısından iyi bir programdır.

---

Öncelikle **prsize1.c** yazdım. Bu, mevcut dizindeki tüm dosyaların boyutunu 'print' eder. Bu, **stat** and **opendir/readdir/closedir**'ın basit bir kullanımınıdır. Dizin **test1**'de bunu test edebilirsiniz. Kendinize temiz bir dizin açın ve aşağıdakileri yapın:

```
UNIX> cp ~huangj/cs360/notes/Prsize/*.c .
UNIX> cp ~huangj/cs360/notes/Prsize/makefile .
UNIX> make
...
UNIX> setenv PRDIR `pwd`
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize1
2074
UNIX> ls -la
drwxr-xr-x  3 huangj          512 Sep 23 10:22 .
drwxr-xr-x  7 huangj       1024 Sep 23 10:37 ..
drwxr-xr-x  2 huangj          512 Sep 23 10:22 d1
-rw-r--r--  1 huangj          11 Sep 23 10:22 f1
-rw-r--r--  1 huangj          15 Sep 23 10:22 f2
UNIX> dc
512 1024 + 512 + 11 + 15 + p
2074
q
UNIX>
```

The "**setenv**" satırı bunu oluşturur ve böylece **prsize1**'i herhangi bir dizinden çağırabilirsiniz. "**ls -l**" ve "**dc**"den görebileceğiniz üzere "**test1**" dizindeki tüm dosyaların boyutunu bir araya toplar. Atacağımız bir sonraki adım mevcut dizindeki tüm erişilebilir dosyaların boyutunu bir araya getirmek olacak. Bunu yapmak için, programı tekrarlanabilir hale getirmeliyiz. Tüm kodları **main()** routine'e eklemek yerine, bir

'function' ile paketleyerek bu 'function' ı çağıracağız. Prsize2.c tam olarak bunu yapıyor. Boyutu bulmak üzere `get_size()`'ı çağırması dışında, `prsize1.c` ile aynı fonksiyonalliteyi sağlar. Henüz 'recursion' söz konusu değil, bu `prsize3.c` için geçerli. `prsize2`'ı test ederseniz `prsize1` ile aynı şeyi yaptığını görebilirsiniz.

```
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize2
2074
UNIX>
```

Şimdi, `prsize2`'ı tekrarlanabilir yapmak istiyoruz. Bir dizin ile karşı karşıya olduğumuzda, bu dizindeki her şeyin boyutunu bulmak isteriz. Bu nedenle tekrarlanabilir şekilde `get_size()`'ı bu dizine çağıracağız. Bu, `prsize3.c`'de yapıldı. Bunu şu dizinde deneyebilirsiniz: `~huangj/cs360/notes/Prsize/test1`

```
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize3
prsize: Too many open files
UNIX>
```

Yani ne oluyor? Bunu kontrol etmek üzere `prsize3a.c`'a bir çıktı (print) açıklaması ekledim. Böylece tekrarlanan çağrılar ne zaman yaptığını görebileceğim.

```
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize3a
Making recursive call on directory .
Making recursive call on directory .
Making recursive call on directory .
Making recursive call on directory .
....
prsize: Too many open files
UNIX>
```

Şimdi neler olduğunu görebilirsiniz. "."daki dosyaları birer birer saydığınızda, "." dosyası ile karşılaşsınız. Bu bir dizindir, bu nedenle onun üzerinde recursive çağrılar yaparsınız. Bu, `opendir()` 'ın başarısız olduğu noktada open file descriptors'ın etkisiz hale gelmesi ile sınırsız bir yinelemeye dönüşür. Bunu düzeltmek için "." dizinine tekrarlanır bir çağrı yapıp yapmadığınızı kontrol etmelisiniz. Ayrıca ".."ı da kontrol etmelisiniz. `Prsize4.c` bu kodu kullanıyor. Şimdi deneyin:

```
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize4
Couldn't stat f3
prsize: No such file or directory
UNIX>
```

Tamam, şimdi problem nedir? Program, dizin `d1`'deki `f3`'ü stat etmeye çalışıyor (bulamadım TR'sini) fakat `d1` dizininde çalışmıyor. Başka bir deyişle, `prsize3`, `~huangj/cs360/notes/Prsize/test1` dizininden çağrılıyor ve `"exists = stat("f3", &buf)"` çağrısını yapıyor. Tabi ki stat -1'e geri dönecek çünkü dizinde `f3` dosyası mevcut değil. Bunun yerine `"d1/f3"`ı aramalıyız. Başka bir deyişle, kodumuzda bir bug var -- `de->d_name` 'i değil, `get_size()` içindeki `fn/de->d_name`'ı arıyor olmalıyız. `Prsize5.c`, bu değişikliğe sebep oluyor.

```
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize5
```

3115

Böylece, bu uygun görünüyor ancak hala yanlış bir şeyler var:

```
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> ls -la
total 5
drwxr-xr-x  3 huangj          512 Sep 23 10:22 .
drwxr-xr-x  7 huangj       1024 Sep 23 10:37 ..
drwxr-xr-x  2 huangj          512 Sep 23 10:22 d1
-rw-r--r--  1 huangj          11 Sep 23 10:22 f1
-rw-r--r--  1 huangj          15 Sep 23 10:22 f2
UNIX> ls -la d1
total 3
drwxr-xr-x  2 huangj          512 Sep 23 10:22 .
drwxr-xr-x  3 huangj          512 Sep 23 10:22 ..
-rw-r--r--  1 huangj          17 Sep 23 10:22 f3
UNIX> dc
512 1024 + 512 + 11 + 15 + 17 + p
2091
512 1024 + 512 + 11 + 15 + 512 + 512 + 17 + p
3115
q
UNIX>
```

Görebileceğiniz üzere **prsize5**, **d1** ve **d1/.**'i farklı iki dosya olarak sayıyor ve her ikisinin boyutlarını toplama ekliyor. **.** ve **d1/..** için de aynı şey geçerli.

Bu bitmeyen bir durum. Daha net olmak için **test2**'ye bakın:

```
UNIX> cd ~huangj/cs360/notes/Prsize/test2
UNIX> ls -la
drwxr-xr-x  2 huangj          512 Sep 23 10:26 .
drwxr-xr-x  7 huangj       1024 Sep 23 10:37 ..
-rw-r--r--  2 huangj          11 Sep 23 10:22 f4
-rw-r--r--  2 huangj          11 Sep 23 10:22 f4-hard-link
UNIX> $PRDIR/prsize5
1558
UNIX> dc
512 1024 + 11 + 11 + p
1558
q
UNIX>
```

**f4** ve **f4-hard-link** dosyaları aynı dosyaya giden linkler. Bununla beraber, **prsize5** onların ayrı olduğunu varsayıyor. Bu nedenle, **prsize** için ihtiyacımız olan, hard linklerin farkına varmak ve onları sadece bir kez saymak.

iki dosya aynı disk dosyasına link ettiğinde nasıl ayrıştırırsınız? Dosya numarasını kullanırsınız. Bu, **buf.st\_ino**.'da bekliyor.

Aynı dosya numaralarını kontrol etmek üzere kullandığımız yöntem, o ana kadar gördüğümüz dosya numaralarının kırmızı-siyah ağacının bakımını yapmak. (bu da saçma oldu konuşalım ☺) Aksi halde, boyuta ekleriz ve dosya numarasını kırmızı-siyah ağaca ekleriz. Dosya numaraları

tamsayı olduğu için kırmızı-siya ağaca giriş yapmak ve modifiye etmek için **jrb\_insert\_int** ve **jrb\_find\_int**'i kullanabiliriz. Kodu **prsize6.c**'de bulabilirsiniz.

```
UNIX> cd ~huangj/cs360/notes/Prsize/test2
UNIX> $PRDIR/prsize6
1547
UNIX> cd ~huangj/cs360/notes/Prsize/test1
UNIX> $PRDIR/prsize6
2091
```

Now, soft links present a small problem. Look at the **test3** directory.

```
UNIX> cd ~huangj/cs360/notes/Prsize/test3
UNIX> ls -la
drwxr-xr-x  2 huangj      512 Sep 23 10:26 .
drwxr-xr-x  7 huangj    1024 Sep 23 10:37 ..
-rw-r--r--  1 huangj     11 Sep 23 10:22 f5
lrwxrwxrwx  1 huangj      2 Sep 23 10:26 f5-soft-link -> f5
lrwxrwxrwx  1 huangj      1 Sep 23 10:24 soft-link-to- -> .
UNIX> $PRDIR/prsize6
Couldn't stat ./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-
./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-
./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-
./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-./soft-link-to-
./soft-link-to-./f5-soft-link
```

Ne oldu? **stat()** kullanıyor olduğumuz andan itibaren, **prsize6** soft linkleri tanımıyor ve bu nedenle daha önce yaşadığımız sınırsız loop problemini tekrar yaşıyoruz. Ne istediğimiz net olabilirdi – linki "." ile çaprazlamak yerine, **prsize**'ın linkin boyutunu kendi kendine saymasını istiyoruz. (**f5-soft-link için 2 bytes** ve **soft-link-to- için 1 byte**). Böylece, **prsize7.c**'te tek yapmamız gereken **stat()** yerine **lstat()** kullanmak. Bu, linkin şaret ettiği dosya yerine soft linkin kendisi hakkında bilgi vermektedir:

```
UNIX> cd ~huangj/cs360/notes/Prsize/test3
UNIX> ls -la
drwxr-xr-x  2 huangj      512 Sep 23 10:26 .
drwxr-xr-x  7 huangj    1024 Sep 23 10:37 ..
-rw-r--r--  1 huangj     11 Sep 23 10:22 f5
lrwxrwxrwx  1 huangj      2 Sep 23 10:26 f5-soft-link -> f5
lrwxrwxrwx  1 huangj      1 Sep 23 10:24 soft-link-to- -> .
UNIX> $PRDIR/prsize7
1550
UNIX> dc
512 1024 + 11 + 2 + 1 + p
1550
UNIX>
```

Son olarak, bu programda bir büyük bug daha var. Open file descriptors ile yapılmalı. **test4** dizininde **prsize7**'yi deneyebilirsiniz:

```
UNIX> cd ~huangj/cs360/notes/Prsize/test4
UNIX> $PRDIR/prsize7
prsize: Too many open files
UNIX>
```

Neler oluyor? Anlayabilmek için, **prsize7a.c**'de **get\_size**'ı edinmek için her çağrıya bir print açıklaması ekledim.

```
UNIX> cd ~huangj/cs360/notes/Prsize/test4
UNIX> $PRDIR/prsize7a
Testing .
Testing ./1
Testing ./1/2
Testing ./1/2/3
Testing ./1/2/3/4
Testing ./1/2/3/4/5
...
Testing
./1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/2
9/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/5
5/56/57/58/59/60/61
prsize: Too many open files
UNIX>
```

**get\_size()**'ı yakalamak için yapılan tekrarlı çağrılar **opendir()** ve **closedir()** çağrıları arasında yapılıyor. Tekrarlı çağrı yaptığımız her seferinde açık dosya numarasına bir tane daha eklemiş oluyoruz. Unix, herhangi bir işlem tarafından kaldırılacak sınırlı sayıda açık dosyaya izin verdiği için, içiçe geçmiş tekrarlı çağrılar yaptığımızda hata mesajı alırız. Bunun çözüm yolu,tekrarlı çağrılar yaptığımızda open files olmamasına dikkat ediyor olmak. Bunu nasıl yapabiliriz? Biz dizindeki dosyaları numaralandırırken tüm dizinleri bir dlist'e ekleriz ve sonrasında dizin dosyasını kapadıktan sonra listeyi çaprazlarız ve tekrarlanır çağrıları yaparız. Dizinleri dlist'e eklediğimizde **astrdup()** yapmamız gerektiğini unutmayın. Neden? Düşünün veya bunu yapmadığımızda neler olabileceğini kendiniz görün ve programı **test5** dizininde çalıştırmayı deneyin.

**prsize8.c**'deki prsize'nin doğru ve güncel versiyonu:

```
UNIX> cd ~huangj/cs360/notes/Prsize/test4
UNIX> $PRDIR/prsize8
33792
UNIX> cd test5
$PRDIR/prsize8
2656
UNIX>
```