

Özyinelemeli klasör dolaşımı (Recursive directory traversal)

Kaynak: <http://web.eecs.utk.edu/~plank/plank/classes/cs360/>

Özyinelemeli klasör dolaşımı

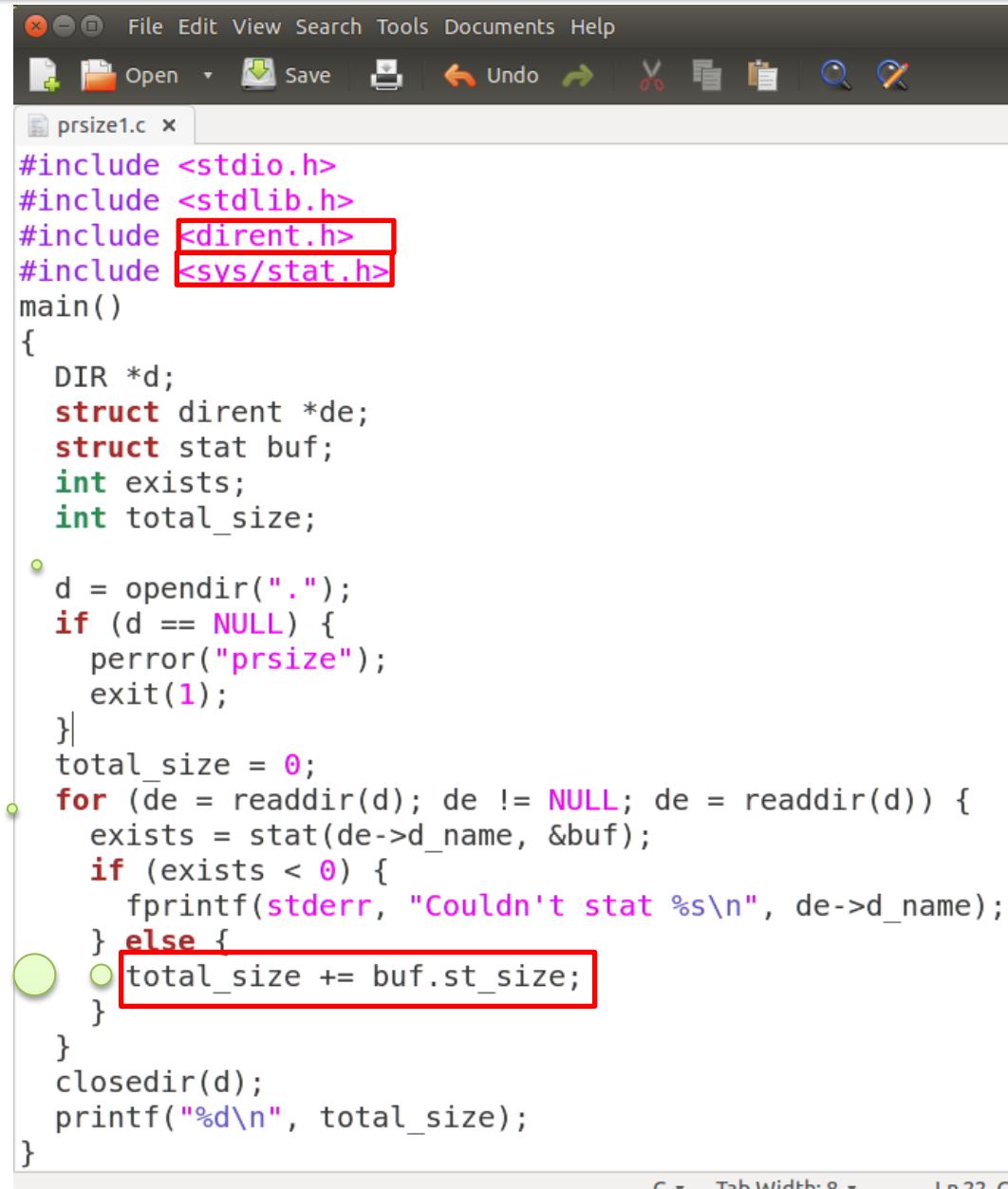
- Bir klasör içinden erişilebilen dosyaların toplam büyüklüğünün ölçülmek için tüm dosyaların boyutlarına bakıp toplam boyutu belirlemek gereklidir.
- opendir/readdir/closedir, stat, recursion, building path names, finding hard links
- Prsize isimli program ile çalışma klasöründeki tüm dosyaların toplam boyutunu hesaplanacaktır.

Örnek 1: toplam dosya boyutu

Çalışma klasörünü aç

Klasördeki dosyaları sırayla oku

Stat sistem çağrıları ile ulaşılan dosya boyutları toplanıyor



```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
#include <sys/stat.h>
main()
{
    DIR *d;
    struct dirent *de;
    struct stat buf;
    int exists;
    int total_size;

    d = opendir(".");
    if (d == NULL) {
        perror("prsize");
        exit(1);
    }
    total_size = 0;
    for (de = readdir(d); de != NULL; de = readdir(d)) {
        exists = stat(de->d_name, &buf);
        if (exists < 0) {
            fprintf(stderr, "Couldn't stat %s\n", de->d_name);
        } else {
            total_size += buf.st_size;
        }
    }
    closedir(d);
    printf("%d\n", total_size);
}
```

Örnek 2: fonksiyon tanımlama (prsizer2.c)

```
prsize2.c x
int get_size(char *fn)
{
    DIR *d;
    struct dirent *de;
    struct stat buf;
    int exists;
    int total_size;

    d = opendir(fn);
    if (d == NULL) {
        perror("prsize");
        exit(1);
    }

    total_size = 0;

    for (de = readdir(d); de != NULL; de = readdir(d)) {
        exists = stat(de->d_name, &buf);
        if (exists < 0) {
            fprintf(stderr, "Couldn't stat %s\n", de->d_name);
        } else {
            total_size += buf.st_size;
        }
    }
    closedir(d);
    return total_size;
}

main()
{
    printf("%d\n", get_size("."));
}
```

Dosya boyutlarını
listeleyen program
fonksiyon olarak
tanımlandı.

Örnek 3: özyinelemeli yaklaşım (prsizer3.c)

- Bir klasörün içindeki alt klasörlerin boyutları getsize fonksiyonunun özyinelemeli çalışması ile sağlanmıştır.
- S_ISDIR() fonksiyonu ile bilgileri okunan klasör mü diye bakılır.
- Klasör ise bu klasörün içindekilerin de boyutunun ölçülmesi gereklidir.
- Ancak bu işlem bir hata ile sonuçlanır.

```
prsize3.c
int get_size(char *fn)
{
    DIR *d;
    struct dirent *de;
    struct stat buf;
    int exists;
    int total_size;

    d = opendir(fn);
    if (d == NULL) {
        perror("prsize");
        exit(1);
    }

    total_size = 0;

    for (de = readdir(d); de != NULL; de = readdir(d)) {
        exists = stat(de->d_name, &buf);
        if (exists < 0) {
            fprintf(stderr, "Couldn't stat %s\n", de->d_name);
        } else {
            total_size += buf.st_size;
        }
        /* Make the recursive call if the file is a directory */
        if (S_ISDIR(buf.st_mode)) {
            total_size += get_size(de->d_name);
        }
    }
    closedir(d);
    return total_size;
}

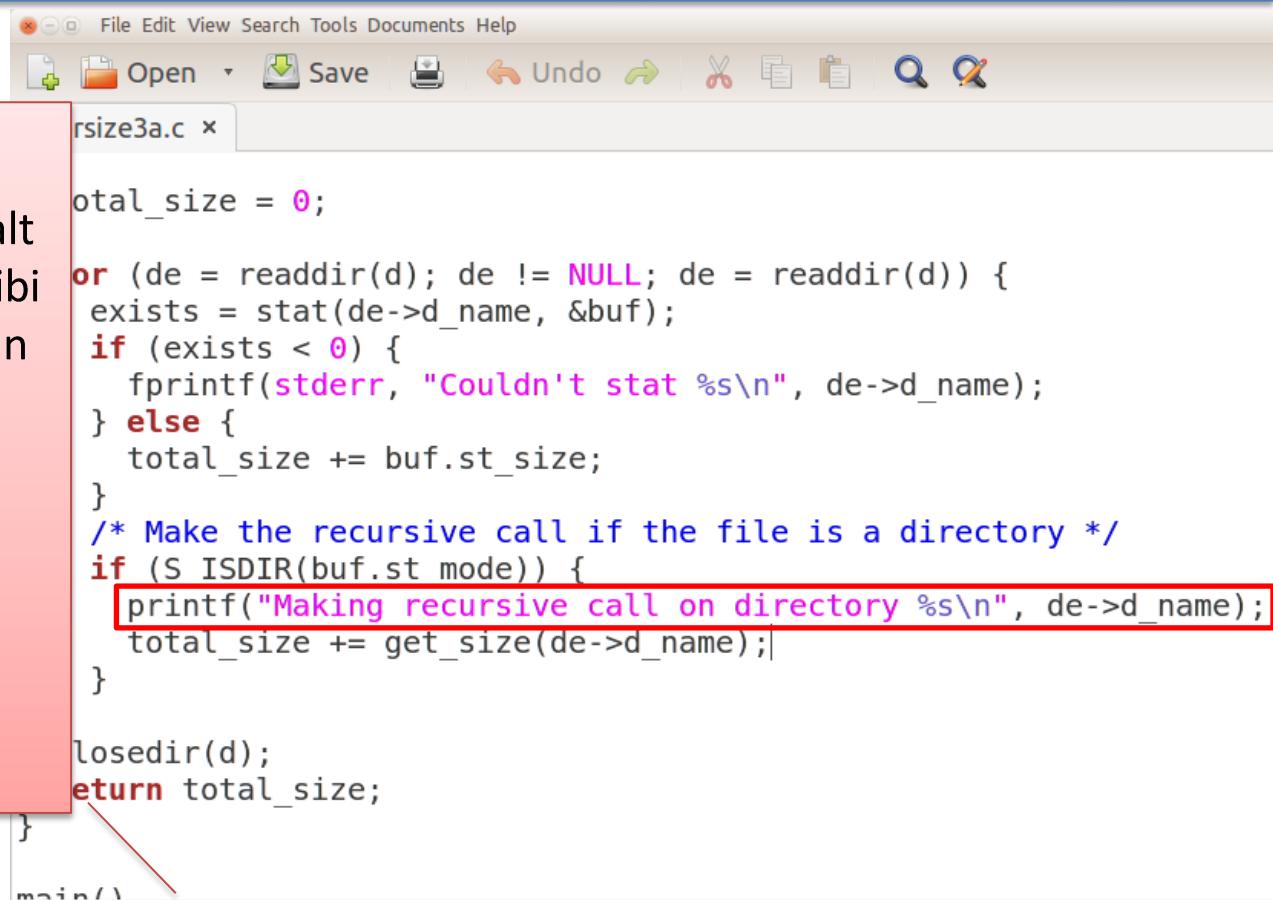
main()
{
    printf("%d\n", get_size("."));
}
```

File Edit View Search Terminal Help

```
test:prsize3
prsize: Too many open files
test:█
```

Örnek 3: hatanın analizi (prsizer 3a.c)

- Önceki program üzerinde, alt klasöre gidilirken yandaki gibi bir açıklama yazılsırsa hatanın nedeni, aşağıdaki çıktıdan anlaşılabilir.
- Program çalıştırıldığında (./prsizer3a) tekrarlı olarak çağrılan klasör; ":" yani klasörün kendisidir.



```
rsizer3a.c x
total_size = 0;

for (de = readdir(d); de != NULL; de = readdir(d)) {
    exists = stat(de->d_name, &buf);
    if (exists < 0) {
        fprintf(stderr, "Couldn't stat %s\n", de->d_name);
    } else {
        total_size += buf.st_size;
    }
    /* Make the recursive call if the file is a directory */
    if (S_ISDIR(buf.st_mode)) {
        printf("Making recursive call on directory %s\n", de->d_name);
        total_size += get_size(de->d_name);
    }
}

closedir(d);
return total_size;
}

main()

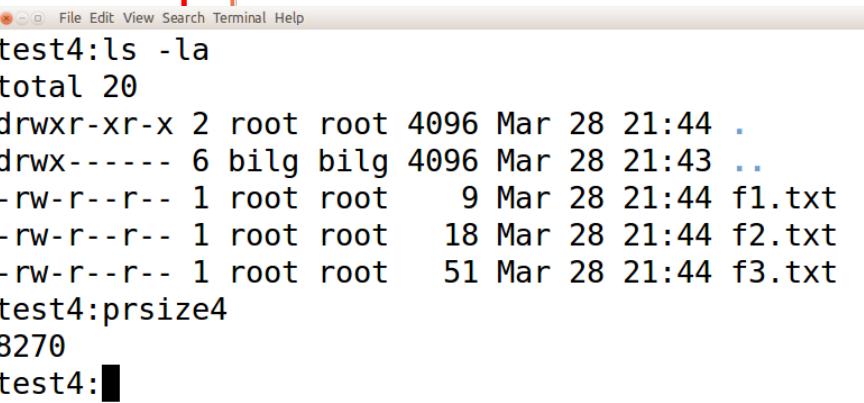
```

```
Making recursive call on directory .
prsize: Too many open files
```

C Tab Width: 8 Ln 38, Col 42 INS

Örnek 4: “.” ve “..” klasörleri (prsizel 4)

- Klasörlerin boyutu hesaplanırken “.” ve “..” özyinelemeli çağrılarında kullanılmazsa program sonsuz döngüye girmez.



```
File Edit View Search Tools Documents Help
Open Save Undo Redo Cut Copy Paste Find Replace
prsize4.c x

for (de = readdir(d); de != NULL; de = readdir(d)) {
    exists = stat(de->d_name, &buf);
    if (exists < 0) {
        fprintf(stderr, "Couldn't stat %s\n", de->d_name);
    } else {
        total_size += buf.st_size;
    }
    /* Make the recursive call if the file is a directory and is not
     . or .. */
    if (S_ISDIR(buf.st_mode) && strcmp(de->d_name, ".") != 0 &&
        strcmp(de->d_name, "..") != 0) {
        total_size += get_size(de->d_name);
    }
}
closedir(d);
return total_size;
}

test4:ls -la
total 20
drwxr-xr-x 2 root root 4096 Mar 28 21:44 .
drwx----- 6 bilg bilg 4096 Mar 28 21:43 ..
-rw-r--r-- 1 root root    9 Mar 28 21:44 f1.txt
-rw-r--r-- 1 root root   18 Mar 28 21:44 f2.txt
-rw-r--r-- 1 root root   51 Mar 28 21:44 f3.txt
test4:prsize4
8270
test4:
```

Örnek 4: : alt klasörler problemi

- Bir önceki programda klasörün içerisinde klasörün bulunması durumunda program aşağıdaki gibi, stat dosyayı bulamaz.

```
File Edit View Search Terminal Help
test4:touch altklasor/deneme.txt
test4:ls
altnasor  f1.txt  f2.txt  f3.txt
test4:prsize4
Couldn't stat deneme.txt
prsize: No such file or directory
test4:cd altnasor
altnasor:ls
deneme.txt
altnasor:|
```

Örnek 5: alt klasörler(prsize5.c)

256 karakter dosya ismi + 2 karakter (/ ve null) + dosya yolu (strlen(fn))

Dosya ismini yoluyla birlikte s değişkenine yazdır.

```
File Edit View Search Tools Documents Help
Open Save Undo Redo Cut Copy Paste Find Replace
prsize5.c x
char *s;
d = opendir(fn);
if (d == NULL) {
    perror("prsize");
    exit(1);
}
s = (char *) malloc(sizeof(char)*(strlen(fn)+258));

for (de = readdir(d); de != NULL; de = readdir(d)) {
    /* Look for fn/de->d_name */
    sprintf(s, "%s/%s", fn, de->d_name);
    exists = stat(s, &buf);
    if (exists < 0) {
        fprintf(stderr, "Couldn't stat %s\n", s);
    } else {
        total_size += buf.st_size;
    }
    if (S_ISDIR(buf.st_mode) && strcmp(de->d_name, ".") != 0 &&
        strcmp(de->d_name, "..") != 0) {
        total_size += get_size(s);
    }
}
closedir(d);

Saving file '/home/bilg/Documents/h... C Tab Width: 8 Ln 20, Col 11 INS
```

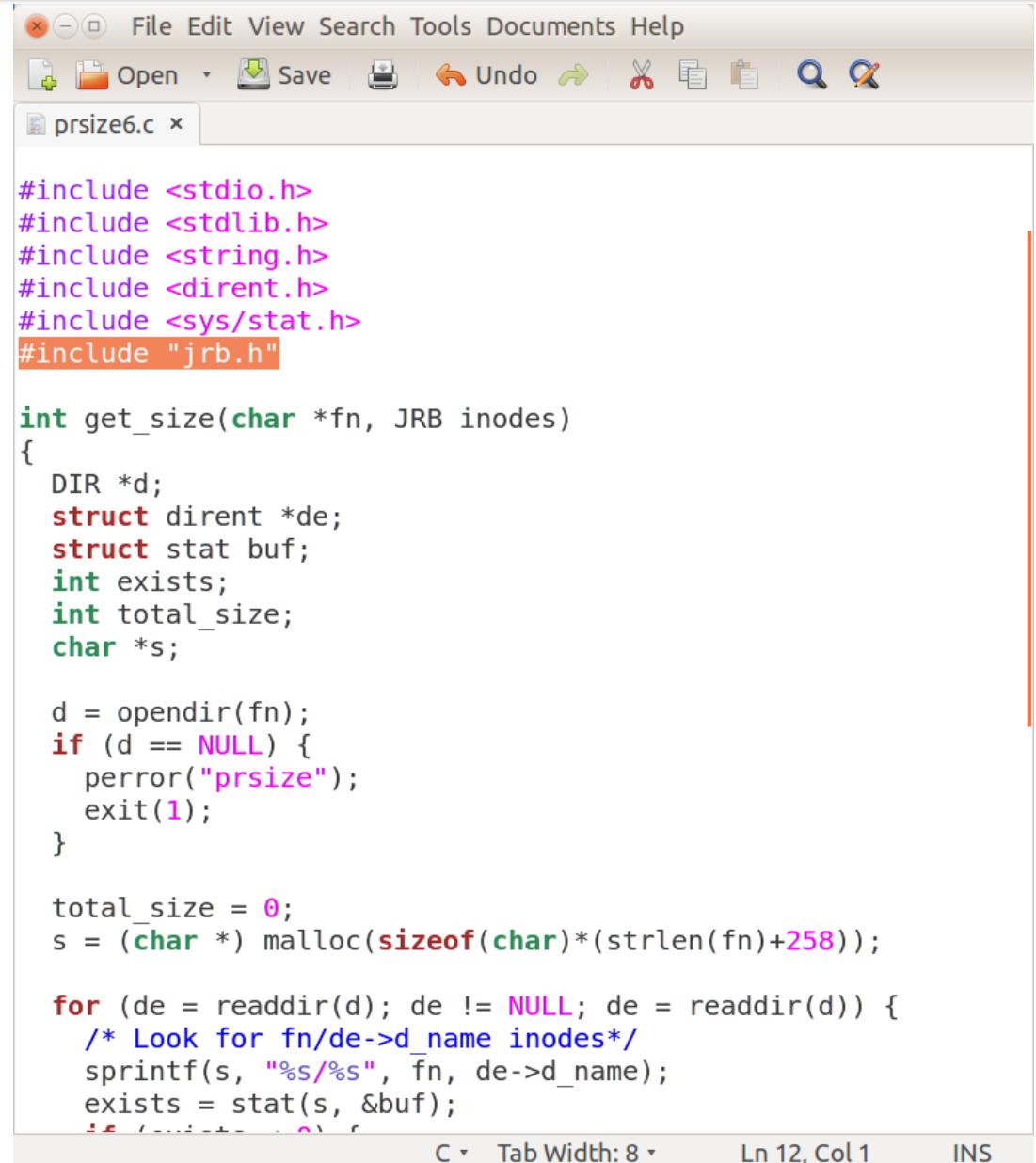
Örnek 5: alt klasörler(prsize5.c)

- prsize5 boyutu 20558 byte hesapladı.
- Ancak burada inode numarası 1840938 ve 1839719 olan klasörler iki kez hesaplandığı için hatalı sonuc elde edilmiştir.

```
File Edit View Search Terminal Help
test4:prsize5
20558
test4:ls -lai
total 24
1840938 drwxr-xr-x 3 root root 4096 Mar 28 22:06 .
1840034 drwx----- 6 bilg bilg 4096 Mar 28 22:18 ..
1839719 drwxr-xr-x 2 root root 4096 Mar 28 22:06 altklasor
1841007 -rw-r--r-- 1 root root    9 Mar 28 21:44 f1.txt
1841008 -rw-r--r-- 1 root root   18 Mar 28 21:44 f2.txt
1841009 -rw-r--r-- 1 root root   51 Mar 28 21:44 f3.txt
test4:ls -lai altklasor
total 8
1839719 drwxr-xr-x 2 root root 4096 Mar 28 22:06 .
1840938 drwxr-xr-x 3 root root 4096 Mar 28 22:06 ..
1840932 -rw-r--r-- 1 root root    0 Mar 28 22:06 deneme.txt
test4:
```

Örnek 6: dosya tekrarının engellenmesi (prsizer 6.c)

- Aynı klasörlerin tekrar kullanımını engellemek için hesaplamaya katılan her dosyanın inode numarası bir listede tutulup kontrol amaçlı kullanılabilir.
- Verilen programda listeyi tutmak için libfdr kütüphanesindeki red-black binary tree kullanılmıştır.
- Bunun için jrb.h programdaki başlık dosyalarında belirtilmiştir.
- Daha önce kaydedilmiş bir inode tekrar ağaca eklenemez



The screenshot shows a code editor window titled "prsizer6.c". The menu bar includes File, Edit, View, Search, Tools, Documents, Help. The toolbar contains icons for Open, Save, Undo, and others. The code itself is written in C and defines a function to calculate the size of a directory. It uses standard library headers like stdio.h, stdlib.h, string.h, dirent.h, sys/stat.h, and jrb.h. The function takes a file name and a JRB structure as parameters. It opens the directory, reads its contents, and checks if the current file has already been processed by looking up its inode number in a red-black tree structure.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include <sys/stat.h>
#include "jrb.h"

int get_size(char *fn, JRB inodes)
{
    DIR *d;
    struct dirent *de;
    struct stat buf;
    int exists;
    int total_size;
    char *s;

    d = opendir(fn);
    if (d == NULL) {
        perror("prsizer");
        exit(1);
    }

    total_size = 0;
    s = (char *) malloc(sizeof(char)*(strlen(fn)+258));

    for (de = readdir(d); de != NULL; de = readdir(d)) {
        /* Look for fn/de->d_name inodes*/
        sprintf(s, "%s/%s", fn, de->d_name);
        exists = stat(s, &buf);
        if (exists && S_ISDIR(buf.st_mode))
            continue;
        if (exists)
            total_size += buf.st_size;
    }
}
```

Örnek 6: dosya tekrarının engellenmesi (prsize 6.c)

```
prsize6.c x
s = (char *) malloc(sizeof(char)*(strlen(fn)+258));

for (de = readdir(d); de != NULL; de = readdir(d)) {
    /* Look for fn/de->d_name inodes*/
    sprintf(s, "%s/%s", fn, de->d_name);
    exists = stat(s, &buf);
    if (exists < 0) {
        fprintf(stderr, "Couldn't stat %s\n", s);
    } else {
        if (jrb_find_int(inodes, buf.st_ino) == NULL) {
            total_size += buf.st_size;
            jrb_insert_int(inodes, buf.st_ino, JNULL);
        }
    }
    if (S_ISDIR(buf.st_mode) && strcmp(de->d_name, ".") != 0 &&
        strcmp(de->d_name, "..") != 0) {
        total_size += get_size(s, inodes);
    }
}
closedir(d);
free(s);
return total_size;
}

main()
{
JRB inodes;
inodes = make_jrb();
printf("%d\n", get_size(".", inodes));
}
```

inode numarasını
ağaçta ara. Yoksa
listeye ekle.

Bu dosyanın tekrar
hesaplamada
kullanılmasını
engellemek için ağaca
key olarak ekle
(value=JNULL)

Örnek 7: soft (sembolik) linklerin tanınması (prsize7.c)

Soft linkler için lstat() sistem çağrımasını kullandık. stat() soft linkleri tanımadığı için soft link için kullanılan alan dikkate alınmıyordu.

stat()
kullanılıyor

lstat()
kullanılıyor

The screenshot shows a terminal window with the following content:

```
prsize7.c x
S = (cchar *) malloc(SIZEOF(cchar)*(strlen(Tn)+258));

for (de = readdir(d); de != NULL; de = readdir(d)) {
    /* Look for fn/de->d_name */
    sprintf(s, "%s/%s", fn, de->d_name);
    exists = lstat(s, &buf);
    if (exists < 0) {
        fprintf(stderr, "Couldn't stat %s\n", s);
    } else {
        if (jrb_find_int(inodes, buf.st_ino) == NULL) {

File Edit View Search Terminal Help
test4:ln -s f1.txt softf1
test4:ls
altklasor f1.txt f2.txt f3.txt softf1
test4:prsize6
12366
test4:prsize7
12372
test4:ls -li
total 16
1839719 drwxr-xr-x 2 root root 4096 Mar 28 22:06 altklasor
1841007 -rw-r--r-- 1 root root 9 Mar 28 21:44 f1.txt
1841008 -rw-r--r-- 1 root root 18 Mar 28 21:44 f2.txt
1841009 -rw-r--r-- 1 root root 51 Mar 28 21:44 f3.txt
1841073 lrwxrwxrwx 1 root root 6 Mar 29 00:36 softf1 -> f1.txt
test4:
```

Örnek 7: bakılan alt dizinlerin yollarının ekrana yazdırılması

```
[bilg dene1]$ ./klasorolustur 10
[bilg dene1]$ prsize7a
Testing .
Testing ./klasor10
Testing ./klasor10/klasor9
Testing ./klasor10/klasor9/klasor8
Testing ./klasor10/klasor9/klasor8/klasor7
Testing ./klasor10/klasor9/klasor8/klasor7/klasor6
Testing ./klasor10/klasor9/klasor8/klasor7/klasor6/klasor5
Testing ./klasor10/klasor9/klasor8/klasor7/klasor6/klasor5/klasor4
Testing ./klasor10/klasor9/klasor8/klasor7/klasor6/klasor5/klasor4/klasor3
Testing ./klasor10/klasor9/klasor8/klasor7/klasor6/klasor5/klasor4/klasor3/klasor2
Testing ./klasor10/klasor9/klasor8/klasor7/klasor6/klasor5/klasor4/klasor3/klasor2/klasor1
49989
[bilg dene1]$
```

- Program çalıştırıldığı klasörden (".") başlayıp tüm alt klasörlere aynı işlemi uyguluyor.
- Bir alt klasöre gidilirken önceki klasör açık kalmasıdır.

```
File Edit View Search Tools Documents Help
Open Save
#!/bin/sh
#klasorolustur
#test amaçlı dizin ve dosya oluşturan script
x=$1
while [ $x -gt 0 ]
do
    mkdir klasor$x
    cd klasor$x
    echo "test dosyası" > test$x.txt
    x=`expr $x - 1`
done
```

Örnek 8: Açık kalan alt klasörlerin kapatılması

- Bir proses içerisinde açılan dosya sayısını azaltmak için alt klasörlere gitmeden önce dosya kapatılabilir.
- Klasör içinde tespit edilen alt klasörlər listeye eklenip, closedir() ile açılan klasör kapatıldıktan sonra alt klasörlere gitme işlemi gerçekleştirilir.
- Programda liste tanımlamak için libfdr ile sağlanan dllist.h kullanılmıştır.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include <sys/stat.h>
#include "jrb.h"
#include "dllist.h"

int get_size(char *fn, JRB inodes)
{
    DIR *d;
    struct dirent *de;
    struct stat buf;
    int exists;
    int total_size;
    char *s;
    Dllist directories, tmp;

    d = opendir(fn);
    if (d == NULL) {
        perror("prsize");
        exit(1);
    }

    total_size = 0;
    s = (char *) malloc(sizeof(char)*(strlen(fn)+258));

    directories = new_dllist();
    for (de = readdir(d); de != NULL; de = readdir(d)) {
        /* Look for fn/de->d name */
        if (strcmp(de->d_name, fn) == 0)
```

Örnek 8: Açık kalan alt klasörlerin kapatılması

Alt klasör
varsayı listeye
ekle

Alt klasörleri
dolaşmadan önce
mevcut klasörü
kapat

Listedeki tüm alt
klasörleri dolaşır
get_size() ile boyutlarını
ölç (öz yinelemeli çağrı)

```
directories = new_dllist();
for (de = readdir(d); de != NULL; de = readdir(d)) {
    /* Look for fn/de->d_name */
    sprintf(s, "%s/%s", fn, de->d_name);
    exists = lstat(s, &buf);
    if (exists < 0) {
        fprintf(stderr, "Couldn't stat %s\n", s);
    } else {
        if (jrb_find_int(inodes, buf.st_ino) == NULL) {
            total_size += buf.st_size;
            jrb_insert_int(inodes, buf.st_ino, JNULL);
        }
    }
    if (S_ISDIR(buf.st_mode) && strcmp(de->d_name, ".") != 0 &&
        strcmp(de->d_name, "..") != 0) {
        dll_append(directories, new_jval_s(strdup(s)));
    }
}
closedir(d);
dll_traverse(tmp, directories) {
    total_size += get_size(tmp->val.s, inodes);
    /* This keeps the program from overgrowing its memory */
    free(tmp->val.s);
}

/* As does this */
free_dllist(directories);
free(s);
return total_size;
}

main()
{
    JRB inodes;
    inodes = make_jrb();
    printf("%d\n", get_size(".", inodes));
```

İşlem bitince
temizle