File Systems

Dr. Fadi Tirkawi

أنواع الملفات

- ماهو الملف؟ هو عبارة عن مجال منطقي من العناويين التي يتم فيها تخزين كتلة من المعلومات
- أنواع الملفات:
  - معلومات
    - رقمي
    - Binary
  - برامج
    - مصدر
    - Object
  - مجلدات

File Attributesخصائص الملفات

- الأسم يتم بشكل قابل للقراءة
- النوع لتحدد نوع الدعم
- المكان مؤشر أين يوجد الملف
- الحجم
- الحماية تحدد من يستطيع قراءة، كتابة، تنفيذ
- الوقت التاريخ و محدد المستخدم

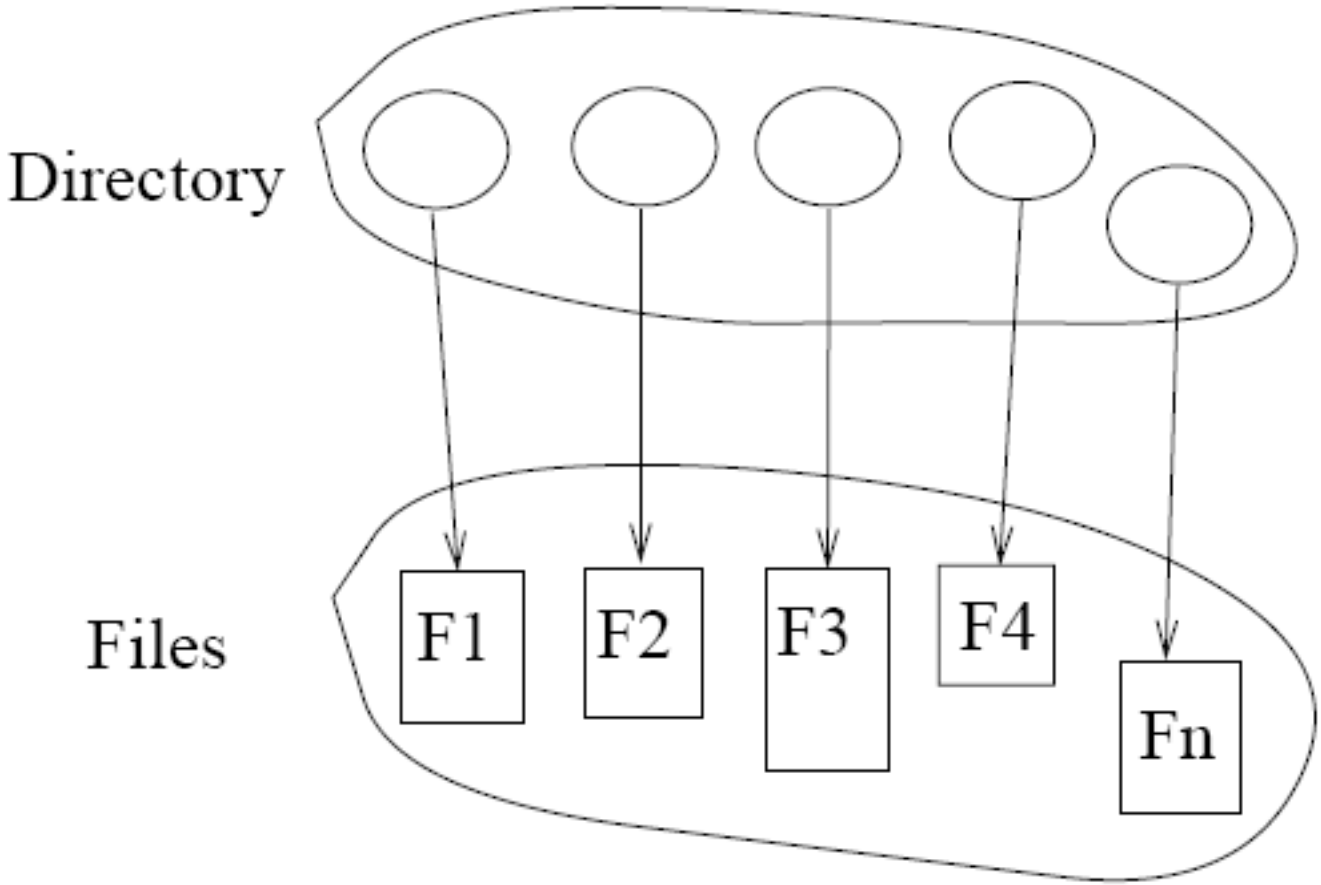
File Attributes

- Name. The symbolic file name is the only information kept in human-readable form,.
- Identifier. This unique tag, usually a umber, identifies the file within the file system; it is the non-human-readable name for the file.
- Type. This information is needed for systems that support different types of files.
- Location. This information is a pointer to a device and to the location of the file on that device.
- Size. The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.
- Protection. Access-control information determines who can do reading, writing, executing, and so on.
- Time, date, and user identification. This information may be kept for creation, last modification, and last use. These data can be useful for protection, security, and usage monitoring.

أنواع الملفات حسب الإمتداد

File type	Usual extension	Function
Executable	exe, com, bin or none	ready-to-run machine-language program
Object	obj, o	compiled, machine language, not linked
Source code	c, p, pas, f77, asm, a	source code in various languages
Batch	bat, sh	commands to the command interpreter
Text	txt, doc	textual data, documents
Word processor	wp, tex, rrf, etc	various word-processor formats
Library	lib, a	libraries of routines
Print or view	ps, dvi, gif	ASCII or binary file
Archive	arc, zip, tar	related files grouped into one file, sometimes compressed

بنية المجلد



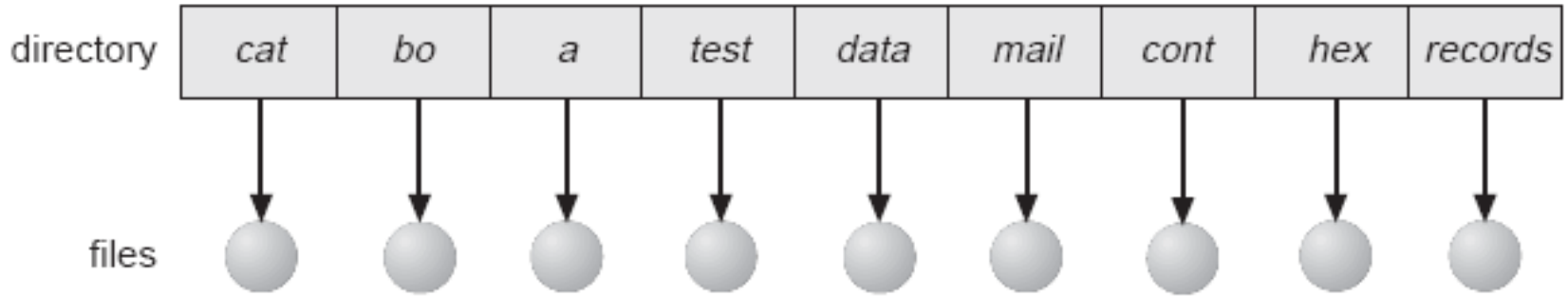
استدعاءات النظام على الملفات

- خلق الملف
- كتابة
- قراءة
- إعادة توضيع المكان
- الحذف
- فتح
- إغلاق

العمليات المنفذة على المجلد

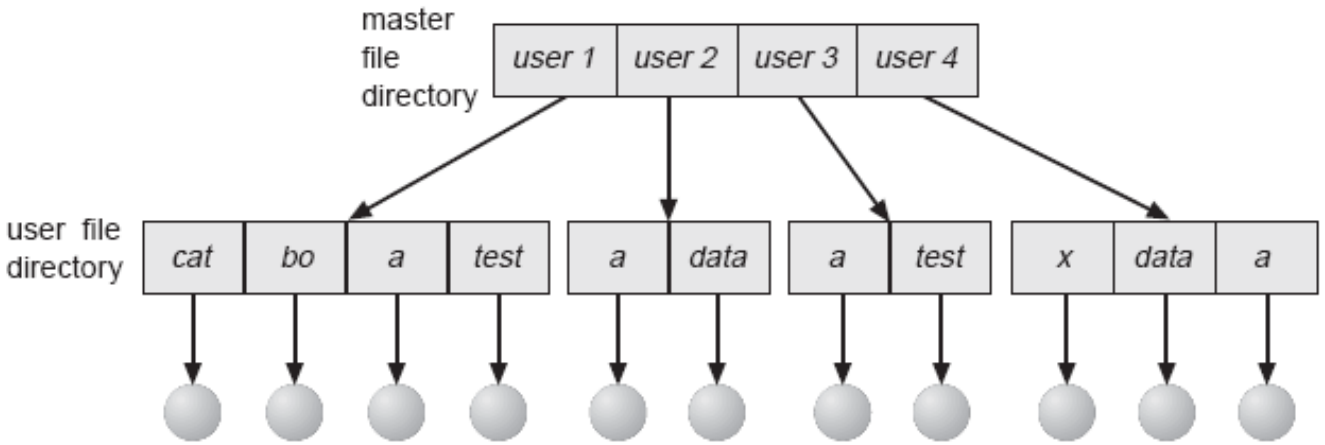
- بحث
- خلق
- حذف
- أظهر الملفات التي في مجلد ما
- أعد التسمية
- تنظيم المجلدات
  - الكفاءة يجب أن نحدد مكان الملف بسرعة و كفاءة
  - التسمية يجب أن تكون ملائمة للمستخدمين

المجلد أحادي المستوى



- مجلد مفرد لجميع المستخدمين
- مشكلة التسمية
- مشكلة المجموعات

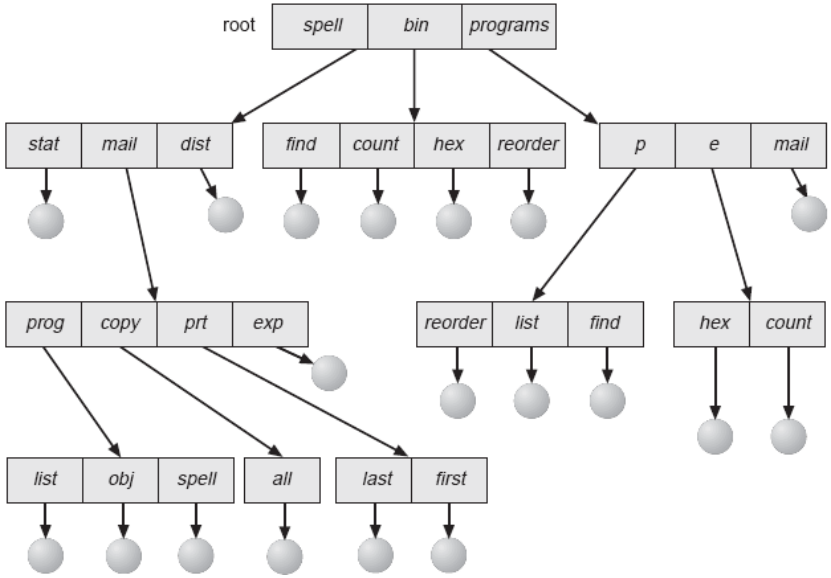
المجلد ثنائي المستوى



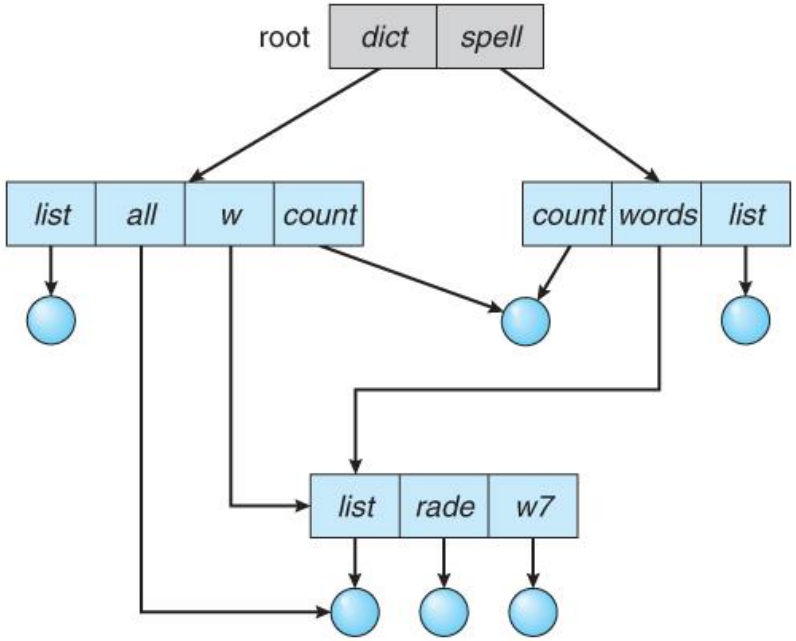
- لكل مستخدم مجلد مستقل
- نستطيع إمتلاك نفس الإسم لمستخدمين مستقلين
- كفاءة في البحث
- لا توجد إمكانية للمجموعات

المجلدات الشجرية

- كفاءة في البحث
- إمكانية المجموعات
- المجلد الحالي هو مجلد العمل
- rm
- mkdir



الغراف الغير دائريAcyclic-Graph Directories



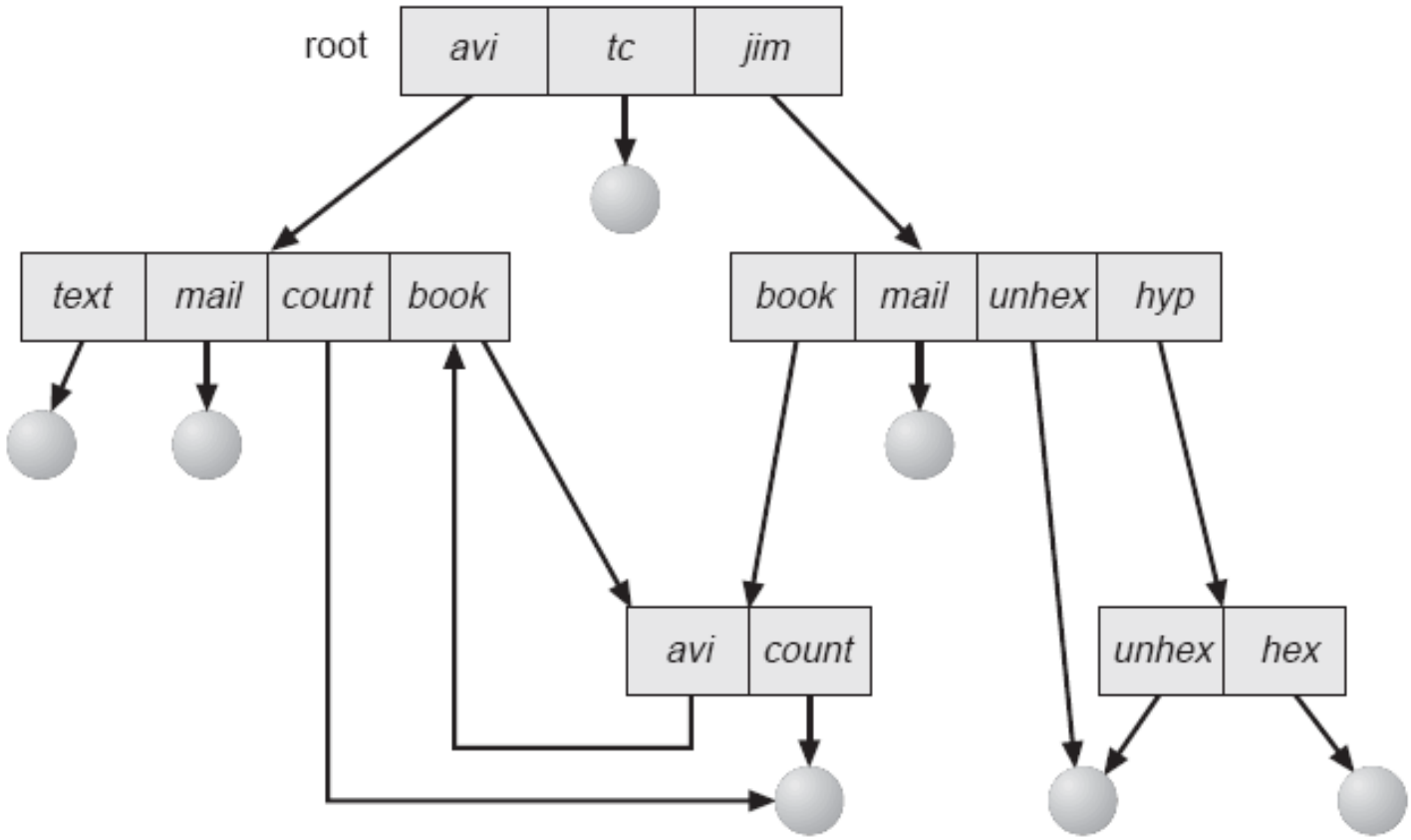
نستطيع مشاركة مجلدات فرعية و ملفات
Share subdirectories or files
نستطيع المشاركة من خلال الوصلاتLINK

- *ln* in UNIX, *fsutil* in Windows for hard links
- *ln –s* in UNIX, shortcuts in Windows for soft links

الميزات

- يمكن إمتلاك أسمين مختلفين aliasing
- هنا لا يوجد حلقات

غراف المجلدات العام



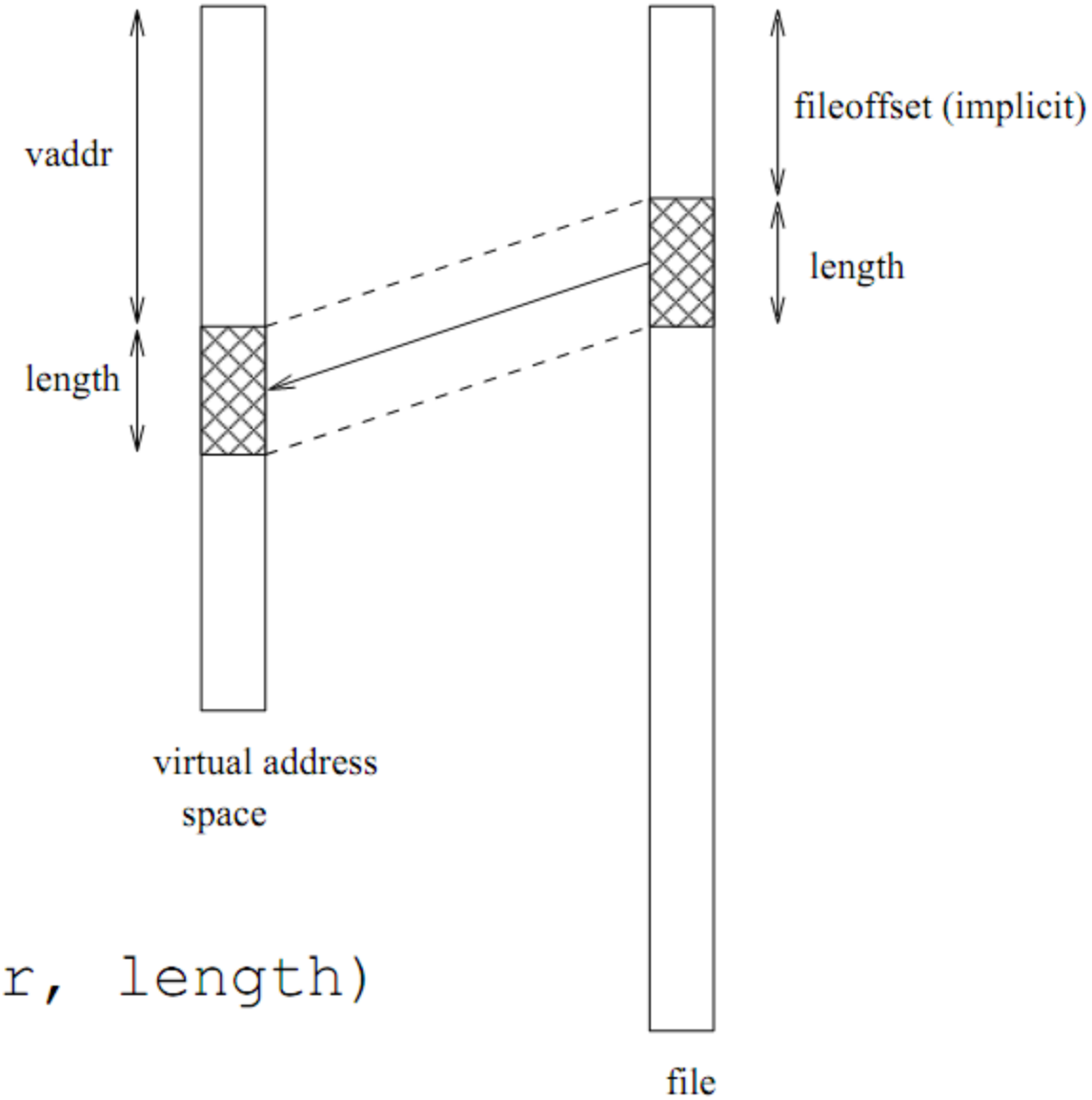
غراف المجلدات العام

- كيف يتم ضمان عدم وجود حلقات
- (One solution is to not follow links in search algorithms.)
- التنظيف Garbage collectoin للتخلص من الأشجار الفرعية.

File Interface
واجهة التخاطب

- open, close
- فتح، إغلاق
- open returns a file identifier (or handle or descriptor), which is used in subsequent operations to identify the file.
- فتح الملف يعيد محدد الملف الذي يستخدم للعمليات على الملف
- read, write
  - must specify which file to read, which part of the file to read, and where to put the data that has been read (similar for write).
- قراءة، كتابة
- Seek
- البحث
- get/set file attributes, e.g., Unix fstat, chmod
- توضيع خصائص الملف

File Read




```
read(fileID, vaddr, length)
```

File Position
موقع الملف

- read and write operations
  - عمليات القراءة و الكتابة
  - start from the current file position يبدأ من موقع الحالي
  - update the current file position يحدث موقع الملف الحالي
- this makes sequential file I/O easy for an application to request هذا يجعل القراء التسلسلية سهلة من اجل التطبيقات
- for non-sequential (random) file I/O, use:
  - seek, to adjust file position before reading or writing
  - a positioned read or write operation, e.g., Unix pread, pwrite: (القراءة بتحديد موقع)
  - pread(fileId,vaddr,length,filePosition)

Sequential File Reading Example (Unix)

```
char buf[512];
int i;
int f = open("myfile",O_RDONLY);
for(i=0; i<100; i++) {
  read(f,(void *)buf,512);
}
close(f);
```

Read the first $100 * 512$ bytes of a file

File Reading Example Using Seek (Unix)

```
char buf[512];
int i;
int f = open("myfile",O_RDONLY);
lseek(f,99*512,SEEK_SET);
for(i=0; i<100; i++) {
  read(f,(void *)buf,512);
  lseek(f,-1024,SEEK_CUR);
}
```

Read the first $100 * 512$ bytes of a file, 512 bytes at a time, in reverse order.

File Reading Example Using Positioned Read

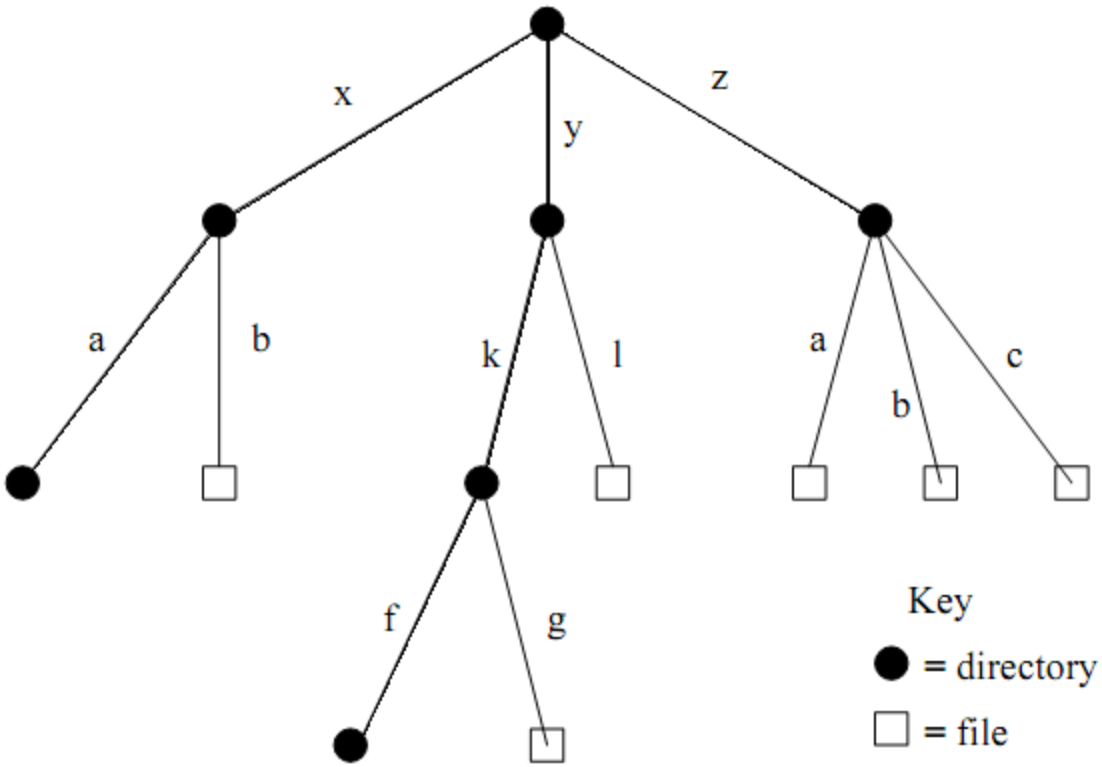
```
char buf[512];
int i;
int f = open("myfile",O_RDONLY);
for(i=0; i<100; i+=2) {
  pread(f,(void *)buf,512,i*512);
}
close(f);
```

Read every second 512 byte chunk of a file, until 50 have been read.

File Names

- أغراض التطبيقات المرئية مثل الملفات و المجلدات تعطى أسماء
- إن نظام الملفات مسؤول عن تخصيص أسماء للأغراض
- مجال الأسماء عادة يكون بنيوي هرمي غالبا شجرة أو غراف موجه غير دائري DAG
- مجال الأسماء namespace يؤمن طرق من أجل مستخدمين و تطبيقات تنظم و تدير المعلومات
- في مجال الاسماء البينوي، الأغراض ربما تكون محدد من خلال pathnames التي توصف الممر من الجذر إلى الغرض الذي نحدده مثال:
- /home/kmsalem/courses/cs350/notes/filesys.ps

Hierarchical Namespace Example

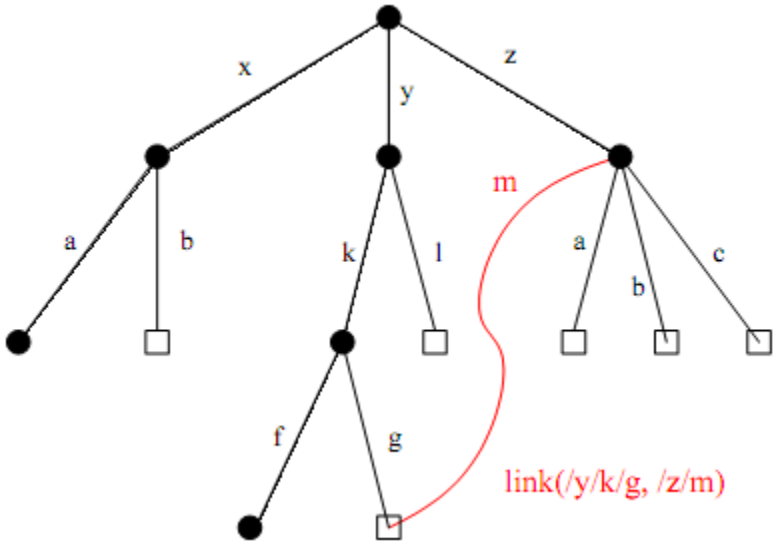


Hard Links

- a hard link is an association between a name and an underlying file (or directory)
- typically, when a file is created, a single link is created to the that file as well (else the file would be difficult to use!)
  - POSIX example: creat(pathname,mode) creates both a new empty file object and a link to that object (using pathname)

POSIX example: `link(oldpath,newpath)`

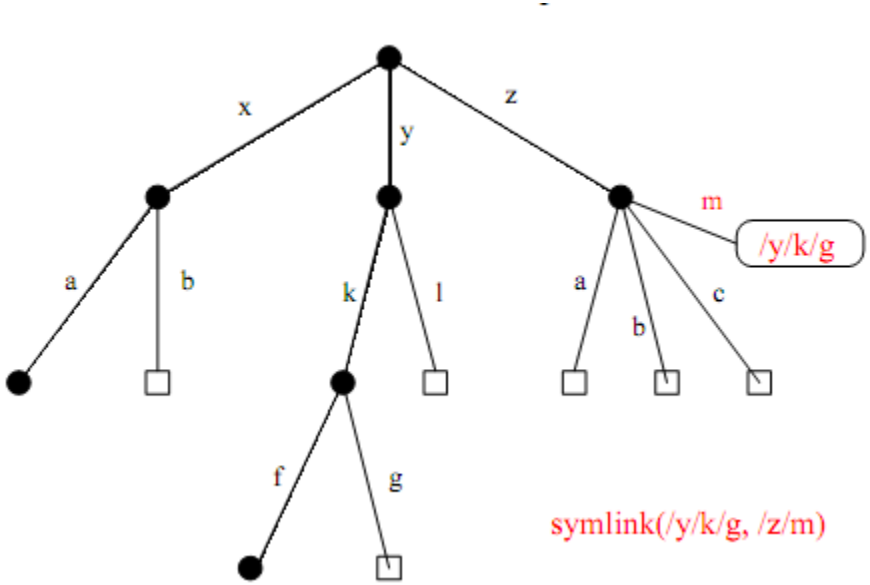
Hard Link Example



Symbolic Links

- a Symbolic link , or soft link, is an association between two names in the file namespace. Think of it is a way of defining a synonym for a filename.
  - symlink(oldpath,newpath) creates a symbolic link from newpath to oldpath, i.e., newpath becomes a synonym for oldpath.

Soft Link Example



`/y/k/g` still has only one hard link after the `symlink` call. A new symlink object records the association between `/z/m` and `/y/k/g`. `open(/z/m)` will now have the same effect as `open(/y/k/g)`.

```
% cat > file1
This is file1.
% ls -li
685844 -rw------- 1 kmsalem kmsalem 15 2008-08-20 file1
% ln file1 link1
% ln -s file1 sym1
% ls -li
685844 -rw------- 2 kmsalem kmsalem 15 2008-08-20 file1
685844 -rw------- 2 kmsalem kmsalem 15 2008-08-20 link1
685845 lrwxrwxrwx 1 kmsalem kmsalem  5 2008-08-20 sym1 -> file1
% cat file1
This is file1.
% cat link1
This is file1.
% cat sym1
This is file1.
```

A file, a hard link, a soft link.

```
% /bin/rm file1
% ls -li
685844 -rw------- 1 kmsalem kmsalem 15 2008-08-20 link1
685845 lrwxrwxrwx 1 kmsalem kmsalem  5 2008-08-20 sym1 -> file1
% cat link1
This is file1.
% cat sym1
cat: sym1: No such file or directory
% cat > file1
This is a brand new file1.
% ls -li
685846 -rw------- 1 kmsalem kmsalem 27 2008-08-20 file1
685844 -rw------- 1 kmsalem kmsalem 15 2008-08-20 link1
685845 lrwxrwxrwx 1 kmsalem kmsalem  5 2008-08-20 sym1 -> file1
% cat link1
This is file1.
% cat sym1
This is a brand new file1.
```

Different behaviour for hard links and soft links.

Storing Files

- Files can be allocated in different ways:
  - Contiguous allocation
    - All bytes together, in order
  - Linked Structure
    - Each block points to the next block
  - Indexed Structure
    - An index block contains pointer to many other blocks

- https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/11_FileSystemInterface.html