

به نام خدا

Unix

فایل درسیستم عامل

آزمایش

۱۳۹۳

ناصر باقری محمود آبادی

یونیکس چیست؟

مغز فیل در
Unix

یونیکس معروفترین سیستم عاملی است که بر روی انواع مختلف کامپیوترها (از میکرو کامپیوترها گرفته تا ابر کامپیوترها) پیاده سازی شده است.

یونیکس سیستم عاملی چند کاربره و چند برنامه ای می باشد و قابلیت های شبکه ای ، پردازش توزیع شده ، بلادرنگ و چند پردازنده ای در نسخه های متعددی از آن پیاده سازی شده است .

یونیکس دارای سیستمی باز بوده و SOURCE نسخه های زیادی از آن به صورت آزاد در اختیار همه افراد قرار دارد .

این سیستم عامل در محیطی تحقیقاتی متولد شده و در دانشگاهها و مراکز آموزشی و پژوهشی به تدریج قابلیت های زیادی به آن اضافه شده است از آنجا که این سیستم عامل به زبان C نوشته شده است قابلیت حمل آن بر روی انواع مختلف کامپیوترها وجود دارد.

لینوکس یک شبه یونیکس آزاد هست (Linux is a Free Unix Like) که در سال ۱۹۸۳ با شروع پروژه گنو کارش رو آغاز کرد. هدف اصلی پروژه گنو یا GNU این بود که یه سیستم عامل آزاد شبیه به یونیکس رو ایجاد کنه. به همین جهت می بینیم که دستورات * و برنامه *ها و فلسفه *ی این دو سیستم عامل یعنی GNU/Linux و Unix بسیار شبیه یکدیگرند.

فایل اقلام داده ای است که بر روی دیسک، یا هر وسیله ای که بشود اطلاعاتی نظیر موسیقی، فیلم، متن، کتاب و نظایر آنرا به صورت الکترونیکی ذخیره کرد است. در حقیقت هر چیزی که بشود روی کامپیوتر ذخیره کرد به شکل فایل می باشد

در لینوکس همه چیز فایل است حتی سخت افزار پیکربندی شده در روی سیستم.



دایرکتوری چیست؟

مغز فایل در
Unix

در یونیکس ها دایرکتوری همان پوشه یا Folder در ویندوز است.

دایرکتوری ها به دو دسته : دایرکتوری ولد و دایرکتوری فرزند دسته بندی می شوند. دایرکتوری فرزند درون دایرکتوری والد قرار می گیرد. در کل هر دایرکتوری که درون دایرکتوری دیگر باشد زیر دایرکتوری یا Subdirectory گویند.

ساختار سلسله مراتبی دایرکتوری ها و زیر دایرکتوری ها تشکیل یک درخت را می دهد. هر درختی یک ریشه دارد اما ریشه درخت های کامپیوتر در بالای درخت قرار دارند. بطور مثال در ویندوز می توانیم سه درخت داشته باشیم (سه درایو) که نام ریشه های آنها C,D,E باشد. اما در یونیکس ها تنها و تنها یک درخت داریم. ریشه این درخت root نام دارد که با نماد / نشان داده می شود. Root بالاترین دایرکتوری است و بقیه دایرکتوری ها زیر این دایرکتوری هستند. برای پیمایش فایل باید از دایرکتوری root یا / شروع کنیم. (دستور tree)



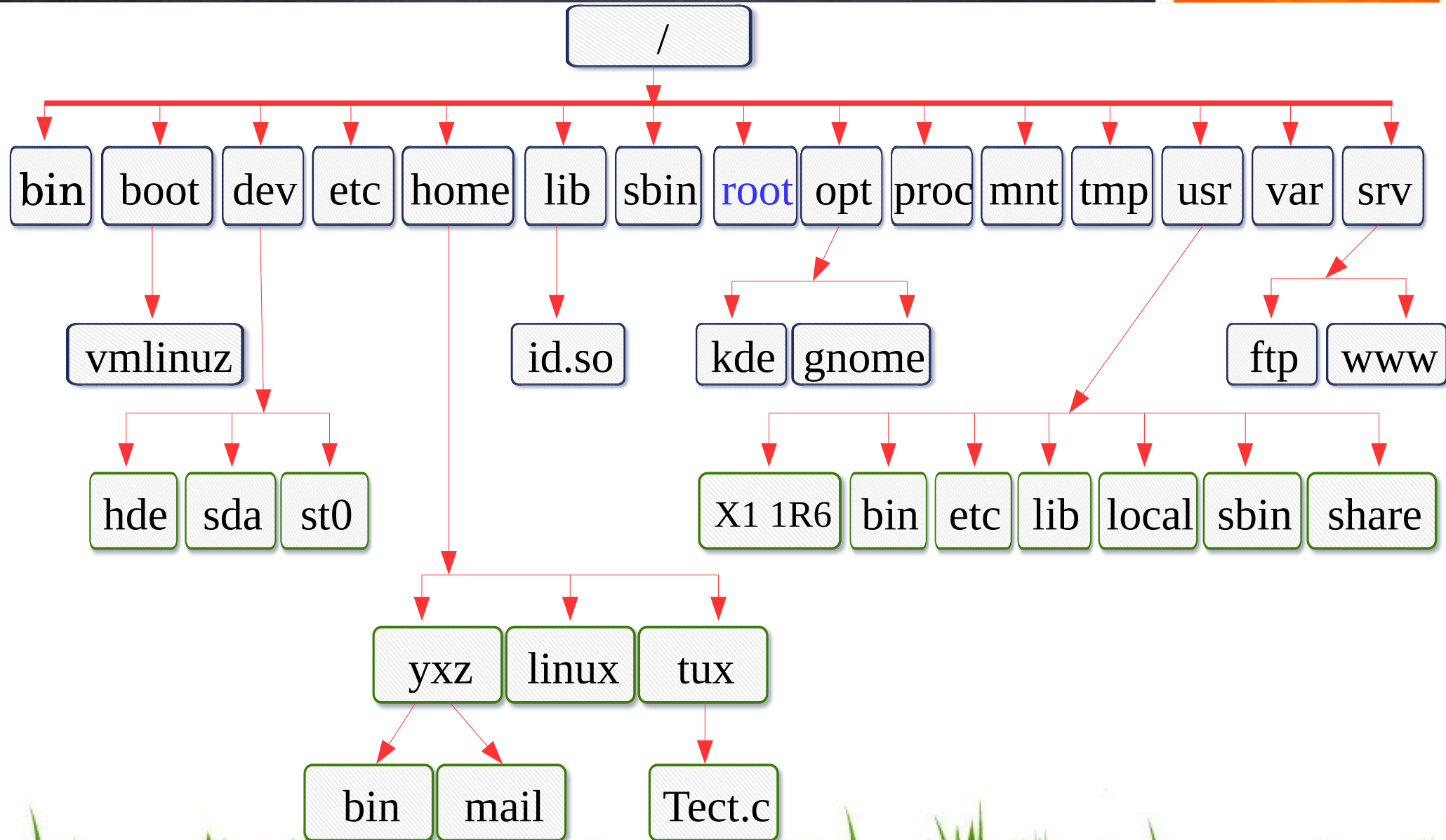
توصیف ساده در مورد سیستمهای لینوکس

اگر بخواهیم یک توصیف ساده در مورد سیستمهای لینوکس ارائه کنیم باید بگوئیم :

- در یک سیستم مبتنی بر یونیکس هر چیزی یک فایل است یا یک فایل نماینده خود را دارد و اگر چیزی یک فایل نباشد حتما یک روند یا پروسه process است .
- در حقیقت در لینوکس فایل‌های مخصوصی وجود دارند که وظایفی بیشتر از یک فایل بر عهده دارند (برای مثال pipe ها و socket ها).
- در یک سیستم گنو/لینوکس هیچ تفاوتی بین فایل و دایرکتوری وجود ندارد ، به این معنی که یک دایرکتوری هم خود یک فایل است که حاوی اسامی تعدادی فایل یا دایرکتوری دیگر است .
- برنامه ها ، سرویسها ، متنها و تصاویر و سایر موارد مشابه و همچنین دستگاههای ورودی و خروجی و عموماً همه ابزارهای سیستمی همگی با یک فایل به سیستم معرفی می شوند .
- حال با این تفکر اگر بخواهیم همگی این فایل‌های متنوع را در یک ساختار منطقی مرتب کرده و نگهداری کنیم باید به آنها بصورت یک ساختار درختی شکل نگاه کنیم که روی هارد دیسک استقرار یافته است (مثل سیستم عامل DOS).

یک نمای کلی از ساختار (سلسله مراتب) فایل (دایرکتوری) در لینوکس

ساختار فایل در
Unix



1- دایرکتوری / یا Root Directory

مسیر اصلی یا ریشه (root)، نقطه آغازین ساختار دایرکتوری در لینوکس است. اینجا همان جایی است که سیستم لینوکس از آن شروع می‌شود. هر فایل و دایرکتوری دیگر، بخشی از این مسیر و یا زیردایرکتوری آن به حساب می‌آید. از آن جایی که به طور پیش فرض تنها زیردایرکتوری در مسیر ریشه وجود دارد، بهترین کار این است که هیچ فایلی را به صورت تنها در این مسیر قرار ندهید

- ذکر این نکته ضروری است که نباید دایرکتوری ریشه را با کاربر root اشتباه گرفت.
- ریشه درخت سیستم فایل لینوکسی و سر آغاز تمامی مسیرهای منتهی به یک دایرکتوری یا فایل در پیمایش سیستم فایل.

`cd /`

- تنها کاربر root مجوز نوشتن بر روی این دایرکتوری (فایل) را دارد.
- این دایرکتوری به عنوان دایرکتوری خانگی کاربر root نیست بلکه دایرکتوری /root دایرکتوری خانگی کاربر root است.

معرفی دایرکتوری‌ها

راهنمای فایل در
Unix

دایرکتوری /bin

شامل تمامی فایل های باینری قابل اجرا (فایل های باینری که مجوز x یا executable را دارند).
دستور های مورد نیاز برای حالت تک کاربره در این دایرکتوری قرار دارند. دستور های زیر این دایرکتوری
تحت مالکیت کاربر و گروه کاربری root هستند اما تمامی کاربران قادر به اجرای آنها هستند. به طور مثال
دستور touch، free یا pwd

دایرکتوری /sbin

همانند دایرکتوری بالا این دایرکتوری نیز شامل فایل های باینری (دستورهای) قابل اجرا است.
دستور های درون این دایرکتوری به منظور مدیریت سیستم به کار رفته و توسط مدیر یا مدیران سیستم
استفاده می شوند. مانند دستور های fdisk, mkfs



معرفی دایرکتوری‌ها

ساختار فایل در
Unix

دایرکتوری /root

این دایرکتوری، دایرکتوری خانگی کاربر root است.

دایرکتوری /home

دیگر کاربران در زیر دایرکتوری /home یک دایرکتوری همنام با نام کاربری خود دارند.

دایرکتوری /etc

این دایرکتوری محل ذخیره فایل‌های پیکربندی سیستم است. به طور مثال فایل پیکربندی سرویس dhcp همچنین در نسخه‌هایی از لینوکس که از فرایند init به منظور کنترل فرایند و سرویس‌ها استفاده می‌کنند، اسکریپت‌های init در زیر دایرکتوری /etc/init.d قرار دارند.



دایرکتوری /proc

در زیر این دایرکتوری به ازای هر فرایند فعال کنونی در سیستم یک زیر دایرکتوری وجود دارد. فایل‌های وضعیتی سیستم و کرنل لینوکس در زیر این دایرکتوری هستند.

دایرکتوری /boot

شامل دایرکتوری‌ها و فایل‌هایی مرتبط با بوت شدن سیستم عامل مانند: فایل تنظیمات گراب - vmlinuz image و فایل‌های دیگر با هسته لینوکس.

دایرکتوری /tmp

برخی از برنامه‌های کاربردی مانند اوراکل یا vmware نیاز به محلی برای ایجاد فایل‌های موقتی دارند که این محل به اندازه‌ای دلخواه به طور پیشفرض دایرکتوری /tmp است. فایل‌های زیر این دایرکتوری پس از reboot شدن سیستم به طور خودکار حذف می‌شوند.

معرفی دایرکتوری‌ها

مغز فایل در
Unix

دایرکتوری `/var`

دایرکتوری برای فایل‌های متغیر یا variable files

در زیر این دایرکتوری فایل‌هایی مانند **log** ها که به سرعت رشد می‌کنند قرار می‌گیرند.

دایرکتوری `/dev`

محل قرارگیری فایل‌های مرتبط به هر device. مانند فایل `sda` که اشاره به اولین دیسک متصل به سیستم می‌کند. مثلا تمام پورت‌ها، پارتیشن‌ها و... در اینجا دارای یک فایل هستند.

دایرکتوری `/lib`

این دایرکتوری شامل فایل‌های کتابخانه‌ای به زبان سی که توسط دستورهای زیر دایرکتوری‌های `/bin` و `/sbin` استفاده می‌شوند.

دایرکتوری `/mnt`

نقاط اتصال به طور معمول در زیر این دایرکتوری متصل یا mount (نرم افزاری) می‌شوند.



معرفی دایرکتوری‌ها

راهنمای فایل در
Unix

دایرکتوری `/media`

محلی برای mount شدن CD/DVD ROM

دایرکتوری `/usr`

دستورهای پایه ای لینوکس مانند `passwd` در زیر دایرکتورهای این دایرکتوری قرار دارند. دوزیر دایرکتوری

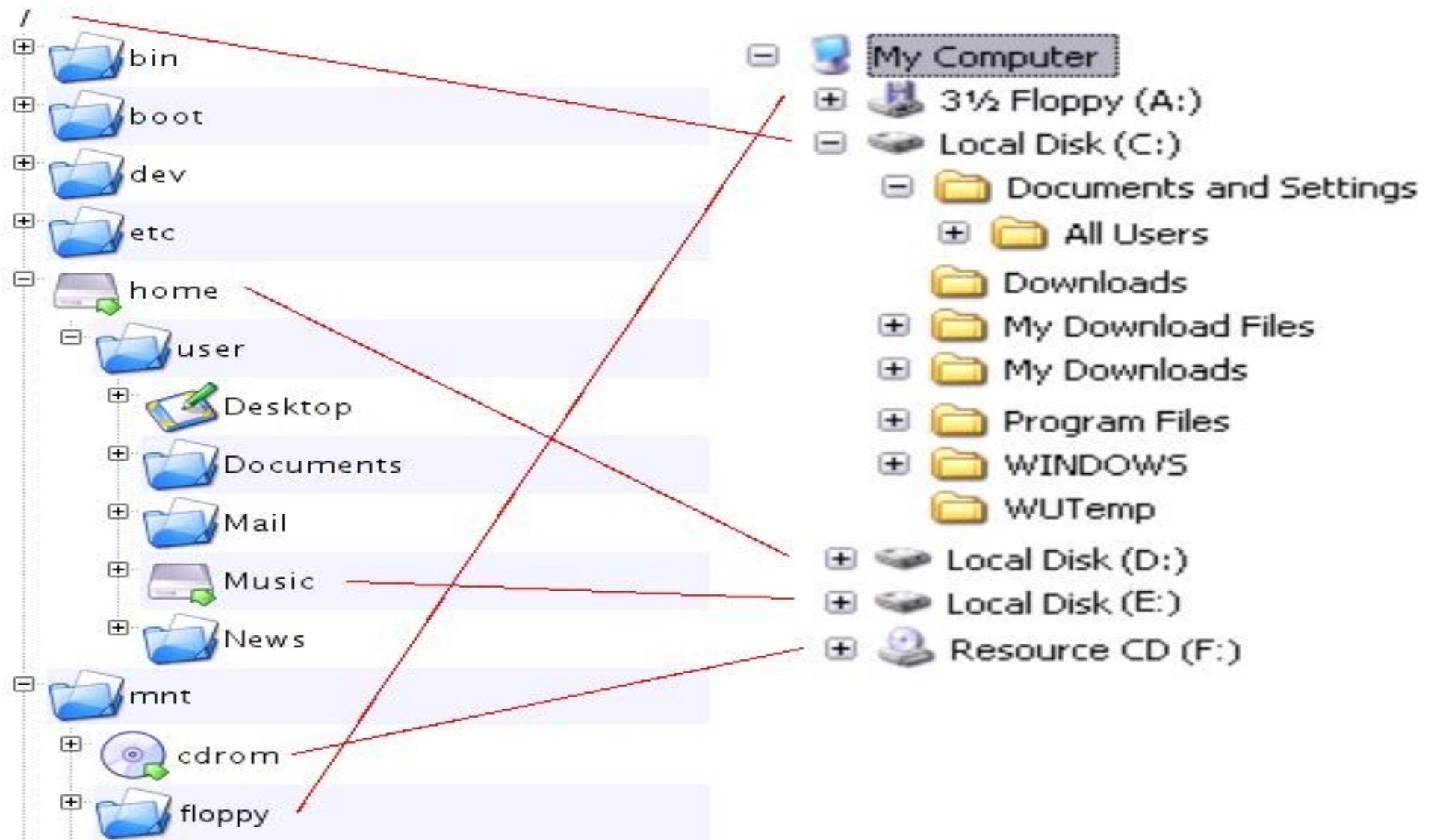
`/usr/bin` و `/usr/sbin`

فایل های راهنما و مستندات زیر دایرکتوری `/usr/share/doc`

دایرکتوری `/srv`

این دایرکتوری برای داده های سرویس های فراهم شده توسط سیستم (Service data) استفاده می شوند.





File Systems: On the left is a typical Linux file system, and on the right is Windows' Explorer. The Linux file system consists of one "tree" with each drive attached to that tree and acting like a folder. The Windows file hierarchy consists of each physical drive having its own separate file system. The lines between the two point to where these drives appear in each graphic.

1) فایل های معمولی (Regular file)

بیشتر فایلها فقط فایل معمولی هستند که حاوی اطلاعات معمولی مثل متن ، کدهای اجرایی برنامه ها ، خروجی و ورودی برنامه های کاربردی دیگر میباشند .

2) فایل های اختصاصی :

دایرکتوری : فایلی حاوی اسامی دیگر فایلها

فایلهای ویژه : حاوی مکانیزم های چگونگی ورود و خروج اطلاعات به کامپیوتر

لینکها : سیستمی را فراهم می کنند که یک فایل یا دایرکتوری در نقاط مختلف ساختار درختی بدون نیاز به

تکرار محتوا قابل مشاهده و در دسترس باشند . (همان short cut ویندوز هست)

Sockets یا Domain نوع بخصوصی از فایل ها که روندهائی که برای کنترل دسترسی به فایل سیستم

در شبکه بکار میروند را فراهم می کنند

pipd : عملی همانند سوکتها انجام می دهند ولی متفاوت با روالی که سوکتها در پیش می گیرند

لینک های Hard

- با لینک های هارد نمی توان دایرکتوری ها را لینک کرد.
- فقط در همان سیستم فایلی که هستند مورد دسترس هستند و در صورت انتقال آنها به یک سیستم فایل دیگر (پارتیشن دیگر) قابل استفاده نیستند.
- اگر فایل اصلی پاک شود، باز هم به آخرین محتوای فایل اصلی دسترسی داریم.

لینک های soft

- می توان یک inode را به چندین نام فایل اختصاص داد.
- برای لینک دادن میان دایرکتوری ها استفاده می شوند.
- می توان در دیگر سیستم فایل ها هم استفاده کرد.
- اگر فایل اصلی پاک شوند دیگر از طریق لینک، به فایل اصلی دسترسی نداریم.

خلاصه انواع فایل

ساختار فایل در
Unix

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special file
S	Socket
p	pipe

مفهوم Inode

راهنمای فایل در
Unix

Inode یا index node که مفهوم اساسی در سیستم عامل های یونیکسی و لینوکس است. هر شی object در سیستم فایل توسط یک inode نشان داده می شود. اشیا یا Object ها چه هستند؟ در یونیکسی ها و لینوکس همه چیز فایل است و همه فایل ها در یونیکس ها و لینوکس دارای خواص و صفات زیر هستند

- **size**. اندازه ای که فایل بر روی دیسک اشغال میکند:
- **Device ID**. مشخص کننده دستگاهی که فایل بر روی آن نگهداری میشود :
- **owner**. هر فایل یک مالک دارد که میتواند کنترل کامل بر روی یک فایل داشته باشد :
- **group**. گروهی که فایل به آن تعلق دارد :
- **permission**. برای جلوگیری از دسترسی افراد غیر مجاز به فایل، برای آن مجوز تعریف میکنند:
- **flags** پرچم، علامتی است که بر روی فایل زده می شود. کاربر یا سیستم میتواند flag هایی را بر روی یک فایل جهت محافظت از آن اعمال کند (مثلاً محدود کردن استفاده و تغییر فایل)

مفهوم Inode

ساختار فایل در
Unix

- access time مشخص کننده آخرین زمان دسترسی به یک فایل است. منظور از دسترسی این است که مثلاً شما بر روی فایل کلیک کرده و محتویات آن را بخوانید. یا یک اسکریپت را از طریق خط فرمان اجرا کنید.
- modification time مشخص کننده آخرین زمان تغییر محتویات یک فایل است. منظور از تغییر کردن این است که شما یک فایل متن را باز کنید و محتویات آن را ویرایش کنید.
- Change time نشان دهنده آخرین زمانی است که اطلاعات inode یک فایل تغییر کند. مثلاً وقتی ما مجوز یک فایل را دستکاری کنیم، این فیلد بروز میشود. این فیلد را با modification time اشتباه نگیرید. اگر محتویات فایل را ویرایش کنیم، modification time بروز میشود. ولی اگر inode، یک فایل را ویرایش کنیم change time بروز میشود.
- inode pointer. یک اشاره گر به بلاکهای از دیسک که فایل به صورت فیزیکی در آنجا نگهداری میشود :
- Link مشخص کننده تعداد :لینکهای سخت. به فایل است
- دقت کنید که 'نام فایل' در inode آن نگهداری نمیشود.

مفهوم Inode

مغز فایل در
Unix

تمامی اطلاعات (صفات) بالا در inode خود مربوط به فایل ذخیره می شوند. بصورت خلاصه inode، فایل و خصوصیات آنرا تعریف می کند. هر inode با یک عدد منحصر به فرد (شیوه خواصی دارد.) درون فایل تعیین می شود. Inode را index number هم می گویند.

چگونه شماره inode را ببینیم:

با استفاده از دستور ls و سوئیچ -i می توان inode هر فایل را دید.

و خروجی آن چیزی شبیه به زیر:

همچنین می توان از دستور stat نیز بصورت زیر استفاده کرد:

و خروجی چیزی شبیه به زیر:

```
Size: 2082          Blocks: 8          IO Block: 4096   regular file
Device: 802h/2050d  Inode: 11537424   Links: 1
Access: (0644/-rw-r--r--)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2014-11-03 09:32:53.601471225 +0330
Modify: 2014-10-09 14:09:02.020211515 +0330
Change: 2014-10-09 14:09:02.076211513 +0330
```


مفهوم Superblock

ساختار فایل در
Unix

هر سیستم فایل دارای Superblock ی است که این SuperBlock شامل اطلاعات زیر است :

- **نوع سیستم فایل** ، سائز فایل سیستم، ویرایش فایل
 - تعداد Inode های موجود
 - آدرس اولین بلاک حاوی اطلاعات
 - عدد کل بلاکهای موجود، تعداد بلاکهای آزاد
 - تعداد Inode های آزاد
 - زمان آخرین بروزرسانی فایل سیستم
 - وضعیت یا Status: اینکه آیا یک فایل سیستم درست Close شده است یا نه
- اگر این اطلاعات از بین بروند، اطلاعات شما از دست می رود (Data Loss) بنابراین لینوکس چندین کپی اضافی از Superblock ها را در هر سیستم فایل نگه داری می کند که در مواقع اضطراری این کپی ها اهمیت دارند.

ترمینال باز کنید و دستور ls -l را بزنید

خروجی:

```
drwxr-xr-x. 2 naser naser 4096 Nov 3 16:18 Desktop
```

=2 تعداد بلاک هایی که در حافظه دایرکتوری Desktop آشغال کرده.

در لینوکس سه حالت یا mode دسترسی وجود دارد که به ترتیب read با علامت r و write با علامت w و اجرایی execute با علامت x نشان داده می شوند. خروجی دستور ls در **۹ ستون** نشان داده می شود که در مورد مجوزها مربوط به ستون نخست است که از **۱۰ کاراکتر** تشکیل شده اند همانطور که گفته شد کاراکتر نخست تعیین کننده نوع فایل که در اینجا d یعنی directory است (طبق جدول) اما ۹ کاراکتر دیگر چگونه هستند. ۹ کاراکتر بعدی در سه دسته سه تایی سازماندهی شده اند که : دسته نخست معرف مجوز مالک و ایجاد کننده فایل است.

دسته دوم معرف مجوز گروه اصلی که مالک فایل عضو آن است

دسته سوم مجوز دیگر افراد است

پس با توجه به این توضیح سه نشان دیگر داریم که : مالک یا user با علامت u و گروه اصلی مالک یا group با علامت g و دیگر کاربران یا other با علامت o در سیستم مجوزدهی لینوکس مشخص می شوند.

مجوز دهی عددی (در سیستم اکتال) :

منظور از سیستم اکتال یعنی اعداد از صفر تا ۷ است و به این صورت تعریف می شود که از راست به چپ x معرف عدد یک و w معرف عدد ۲ و r معرف عدد ۴ است که مجموع این سه عدد بیشتر از ۷ نخواهد شد اما چگونه از آنها استفاده کنیم؟ با توجه به مثال بالا خط زیر را در نظر بگیرید :

$$rwxr-x = 755$$

نتیجه گیری :

- ۱ - دسته قرمز رنگ تعیین کننده مجوز کاربر یا مالک است که اگر سه عدد را با هم جمع کنیم عدد ۷ حاصل می شود و چون هر سه مجوز را دارد یعنی هم می تواند فایل یا دایرکتوری را بخواند و هم در فایل یا دایرکتوری بنویسد و هم آنرا بعنوان یک فایل اجرایی ببیند.
- ۲ - دسته دوم (سبز) می گوید که اعضای گروهی که مالک نیز عضو آن گروه است مجوز خواند و اجرایی بودن فایل را دارند اما نمی توانند بنویسند و اگر اعداد را در دسته دوم از چپ به راست با هم جمع کنیم : $4+1=5$ خواهد بود
- ۳ - برای دیگر کاربران با توجه به دو نمونه بالا عدد ۵ را داریم.
- ۴ - در دو دسته دوم یک علامت - آمده. این علامت هرگاه یکی از مجوز های r , w و یا x را نداشته باشیم جای آنها می آید.
- ۵ - در نهایت مجوز دایرکتوری Desktop (در مثال خط اول) از سمت چپ به راست عدد ۷۵۵ خواهد بود.

مهم :

تا اینجا فهمیدیم که

ساختار فایل (سلسله مراتب فایل)

به صورت درختی است

و با سه مفهوم

superblock - Inode مجوز -

آشنا شدیم

File system

ساختار فایل در
Unix

تعریف File System در کتب علمی به این صورت است: File System یا FS ، لایه‌ای است در هسته سیستم عامل (یا کرنل Kernel) که وظیفه آن، کار با فایل هاست.

پس در کل، "File System" تعیین کننده نوع ذخیره سازی، نوع دسترسی و سازماندهی داده‌ها بر روی هارد دیسک است.

به عبارت دیگر، یک "File System" یا "سیستم پرونده"، یک پایگاه داده تک منظوره برای ذخیره، سازماندهی، دستکاری و بازیابی داده‌ها است.

همچنین توسط فایل سیستم هست که میتوانیم از طریق یک دیسک سیستم عامل خودمان را بوت کنیم

همه ی File System یکسان نیستند و هر کدام از آنها، روش های گوناگونی برای سازماندهی داده ها دارند. برخی فایل سیستم ها از بقیه سریعتر هستند، بعضی نیز ویژگی های امنیتی بیشتری دارند، تعدادی از فضای زیاد برای ذخیره سازی اطلاعات پشتیبانی می کنند در حالیکه برخی دیگر فقط با فضاهای ذخیره سازی بسیار کوچک سازگاری دارند. بعضی از فایل سیستم ها در برابر فایل های مخرب و ویروسها بسیار مقاوم هستند، در حالیکه برخی دیگر سایر ویژگی های یاد شده را فدای ارائه ی سرعت دسترسی بیشتر به کاربران می نمایند.

1- Disk File system

۱- DFS یا Disk File System که شامل موارد زیر است

- ext2 یا second extended file system. در لینوکس استفاده شده است :
- ext3 یا third extended file system. در لینوکس استفاده شده است :
- ext4 یا four extended file system. در لینوکس استفاده شده است :
- FAT توسط مایکروسافت توسعه یافته و توسط DOS تا Win 98 استفاده شده است. و سپس Fat32 و NTFS. نیز ظهور کردند
- Reise file system یا Reise FS
- Jfs
- Xfs
-



- **Ext3: Ext2 :Ext4 = (2-3-4)Extended File system**

حاوی مفاهیمی نظیر بلاک ها، آینودها(Inode) و پوشه ها (directory) می باشد.

Ext3: Ext2 :Ext4 بهبود یافته با قابلیت های ژورنالینگ است. ژورنالینگ امکان بازیافت سریع سیستم فایل را به همراه

پشتیبانی از (POSIX ACL (Access Control List می دهد

- journal هست پس خاصیت Data Recovery دارد.

- بسیار قدرتمند است.

- سرعت خیلی خوبی دارد.

- قابلیت دسترسی خیلی خوبی در سیستم عامل به ما می دهد.

- خاصیت Data Integrity دارد. (تمامیت)

- میتواند فایل ها را تا ۲ ترا بایت ساپورت کند.

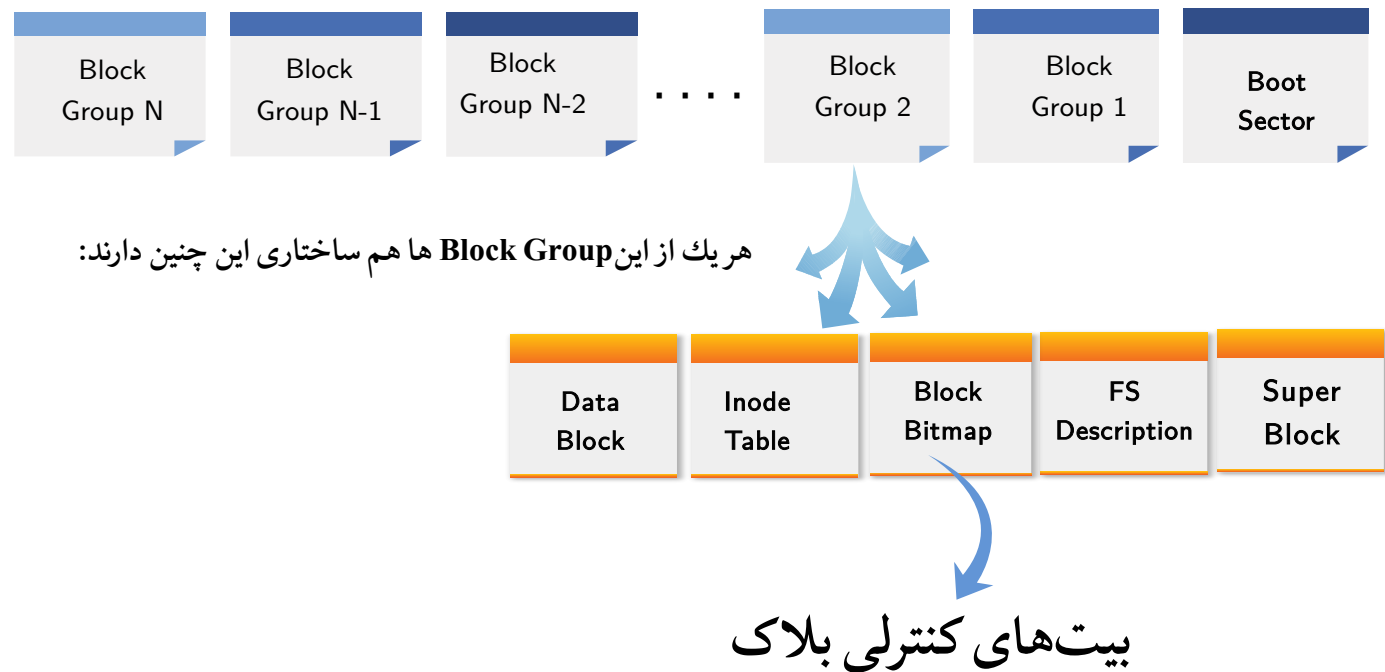
- خاصیت Defragmentation دارد و به صورت اتوماتیک و هوشمند دیسک را دیفراگ میکند که سرعت هارد افزایش

میابد

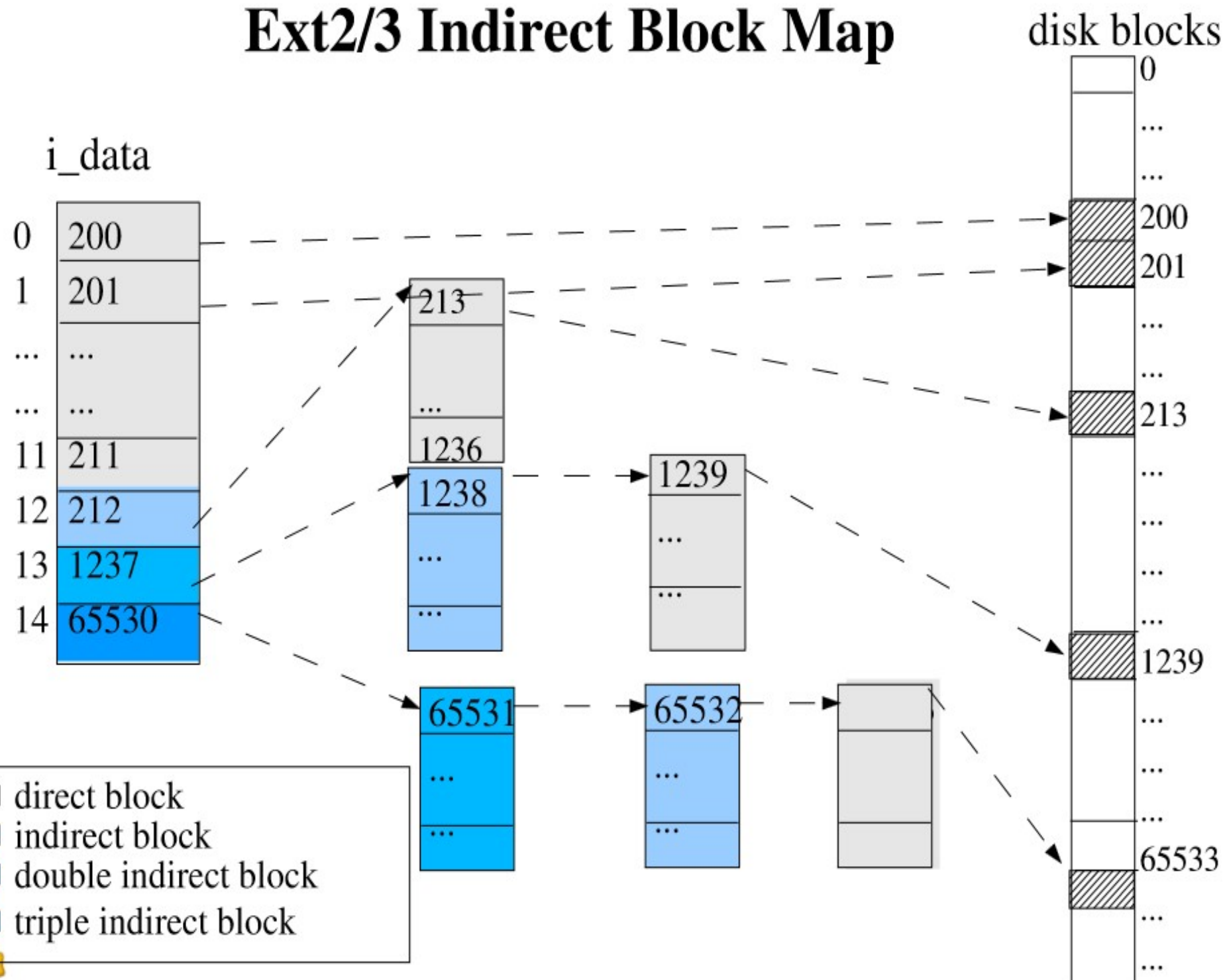
ساختار فیزیکی EXT

ساختار فایل در
Unix

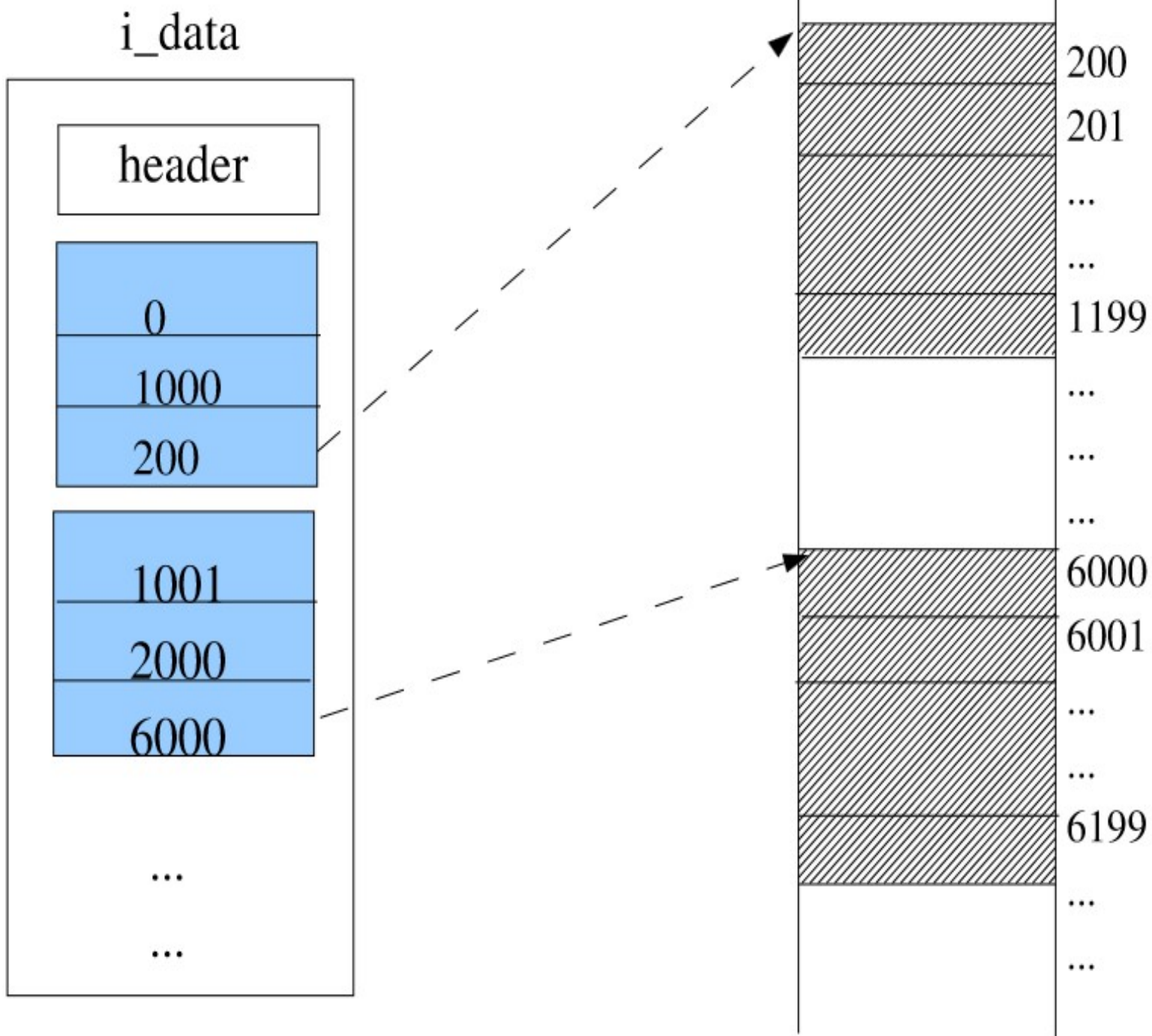
سیستم فایل ext لینوکس ساختار فیزیکی همانند سیستم فایل BSD دارد. بدین صورت که حافظه سیستم فایل تماماً به Block Group ها تقسیم می شود. این بلوک ها در اندازه های 1K، 2K، 4K قرار می گیرند و هر بلوک برای یک سری اطلاعات با کاربردی خاص استفاده می شود. ساختار حافظه فیزیکی سیستم فایل ext به این شکل است:



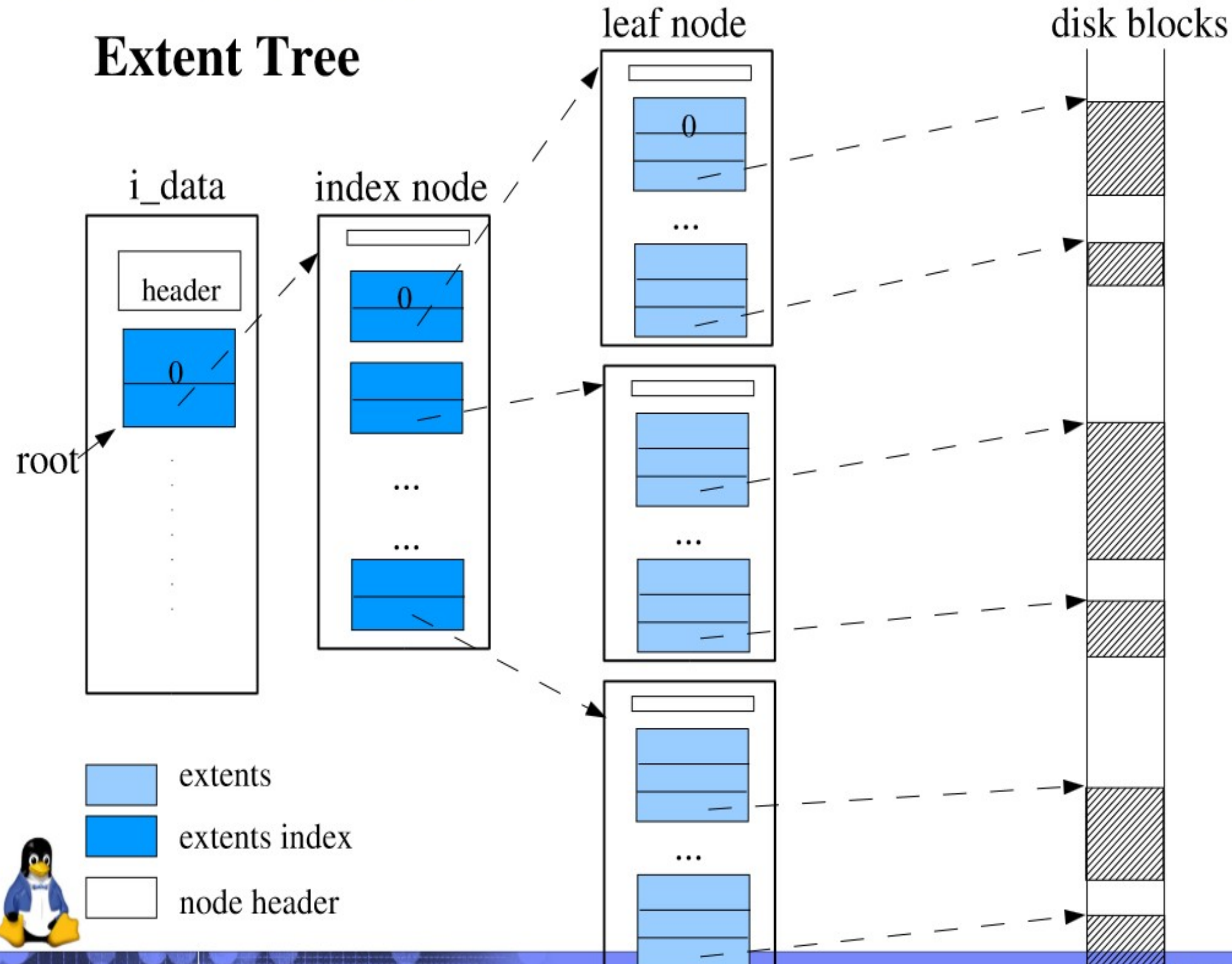
Ext2/3 Indirect Block Map



Extent Map



Extent Tree



DFS-(Reise FS)

مختصر فایل در
Unix

Iso fs (iso9660)

توسط سی دی رام ها استفاده می شود.

Sys fs

یک سیستم فایل مبتنی بر رم که از ابتدا مبتنی بر ram fs بود. برای بارگذاری اشیاء کرنل کاربرد دارد تا کاربر نهایی بتواند از آن به سهولت استفاده کند.

:Proc fs

مثل یک رابط با ساختمان داده داخلی کرنل عمل میکند. برای گرفتن اطلاعاتی راجع به سیستم و تغییر پارامترهای خاص کرنل در زمان اجرا با دستور sysctl استفاده می شود.

به عنوان مثال با دستور زیر می توانید اطلاعات پردازنده را مشاهده کنید:

```
Cat /proc/cpuinfo
```



DFS

ساختار فایل در
Unix

شما با هریک از دو دستور زیر می توانید فایل سیستم های نصب شده روی سیستمتان را ببینید:

Cd /

Cd proc

Ls

Cat filesystem

XFS

ساختار فایل در
Unix

زمانی که سیلیکون گرافیک در سال‌های ۱۹۹۰ نیاز به یک فایل سیستم قدرتمند با بازدهی بالا داشت شروع به توسعه‌ی سیستم فایلی به نام XFS کرد. این سیستم فایل با هدف بالا بردن حجم دیسک سخت و نیز پهنای باند داده‌ها و تبادل موازی برای استفاده در نرم‌افزارهای فیلم و صدا و پایگاه داده‌های بزرگ توسعه یافت. از مزایای این فایل سیستم بازیابی سریع اطلاعات در صورت صدمه دیدن دیسک سخت و پشتیبانی از فایل‌های حجیم و نیز دایرکتوری‌های حاوی فایل‌های بسیار و بازدهی بسیار خوبش در چنین محیط‌هایی است. در حال حاضر XFS با تغییراتی در اختیار جامعه‌ی اوپن سورس قرار گرفته و برای لینوکس نیز بازنویسی شده و برای نسخه‌ی 2.4 و 2.6 هسته‌ی لینوکس موجود است. البته نسخه‌ی لینوکسی هنوز تماماً کامل نیست و برخی قابلیت‌ها مانند loop-mounting هنوز با مشکلاتی همراه است.



از مشکلاتی که در حال حاضر XFS با آن‌ها مواجه است ممکن نبودن **resize** یا تغییر سایز به صورت **on-line** است و برای تغییر سایز باید از اطلاعات پشتیبان بگیرید و پس از تغییر سایز اطلاعات را بازیابی کنید و یا برای استفاده از Quota باید هسته‌ی لینوکس را مجدداً کامپایل کنید. اگر از بوت لودر LILO با گزینه‌ی بوت از بوت سکتور استفاده می‌کنید XFS همراه LILO کار نخواهد کرد. البته با GRUB نسخه‌های بالاتر از 0.99 می‌توانید از XFS استفاده کنید اما مشکلات و رفتار غیر معمول نیز در این حالت گزارش شده است.

اما قابلیت‌های جدید نیز در این فایل سیستم وجود دارد برای نمونه این فایل سیستم یک فایل سیستم ۶۴ بیتی است، بنابراین می‌توانید فایل‌هایی با حجم میلیون‌ها ترابایت داشته باشید که این مقدار بسیار بزرگتر از فایل سیستم‌هایی است که امروزه استفاده می‌شوند. از طرفی این فایل سیستم سرعت بسیار مطلوبی دارد. چیزی حدود هفت گیگابایت در ثانیه (7GB/sec) که این سرعت را مدیون مدل **B+Tree** است که مدل لیست کردن و دسترسی به فایل‌هاست. در سیستم فایل‌های FFS مانند BSD یا Ext2fs از مدل لیست کردن برای دسترسی به فایل‌ها استفاده می‌شود اما در این فایل سیستم از مدل بسیار پیچیده‌ی B+Tree استفاده می‌شود. اما چطور میلیون‌ها ترابایت؟! از دید ریاضی بسیار ساده است:

$$9 \times 10^{18} = 9 \text{ exabytes} \quad 9 = 63^2$$

فایل سیستم JFS

این فایل سیستم که نامش "Journaling File System" است توسط IBM و برای استفاده در سرورهای Enterprise توسعه یافته است. این فایل سیستم برای سرورهایی که ترافیک بالایی دارند و نیاز به فایل سیستم برای مدیریت چنین ترافیکی دارند ساخته شده است. در حال حاضر از این سیستم فایل در لینوکس (هسته ۲.۶) پشتیبانی می شود اما نسخه ی لینوکسی آن هنوز کاملاً برای لینوکس بهینه سازی نشده است. (هنوز کاملاً port نشده)

فایل سیستم JFS از تکنیک "database journaling" برای کارهای خوداستفاده می کند که باعث افزایش سرعت و پاسخ گویی می شود. JFS نیز یک فایل سیستم ۶۴ بیتی است و پارتیشن هایی با حداقل سایز ۱۶ مگابایت (16MB) استفاده می کند. حداکثر سایز کاملاً به سایز بلاک های سیستم بستگی دارد که در حالتی که بلاک ها ۵۱۲ بیتی باشند مقدار بزرگترین فایل برابر ۵۱۲ ترابایت (512TB) و در حالتی که سایز بلاک ها ۴ کیلوبایت باشد حداکثر سایز فایل سیستم ۴ پتابایت (4Peta bytes) خواهد بود

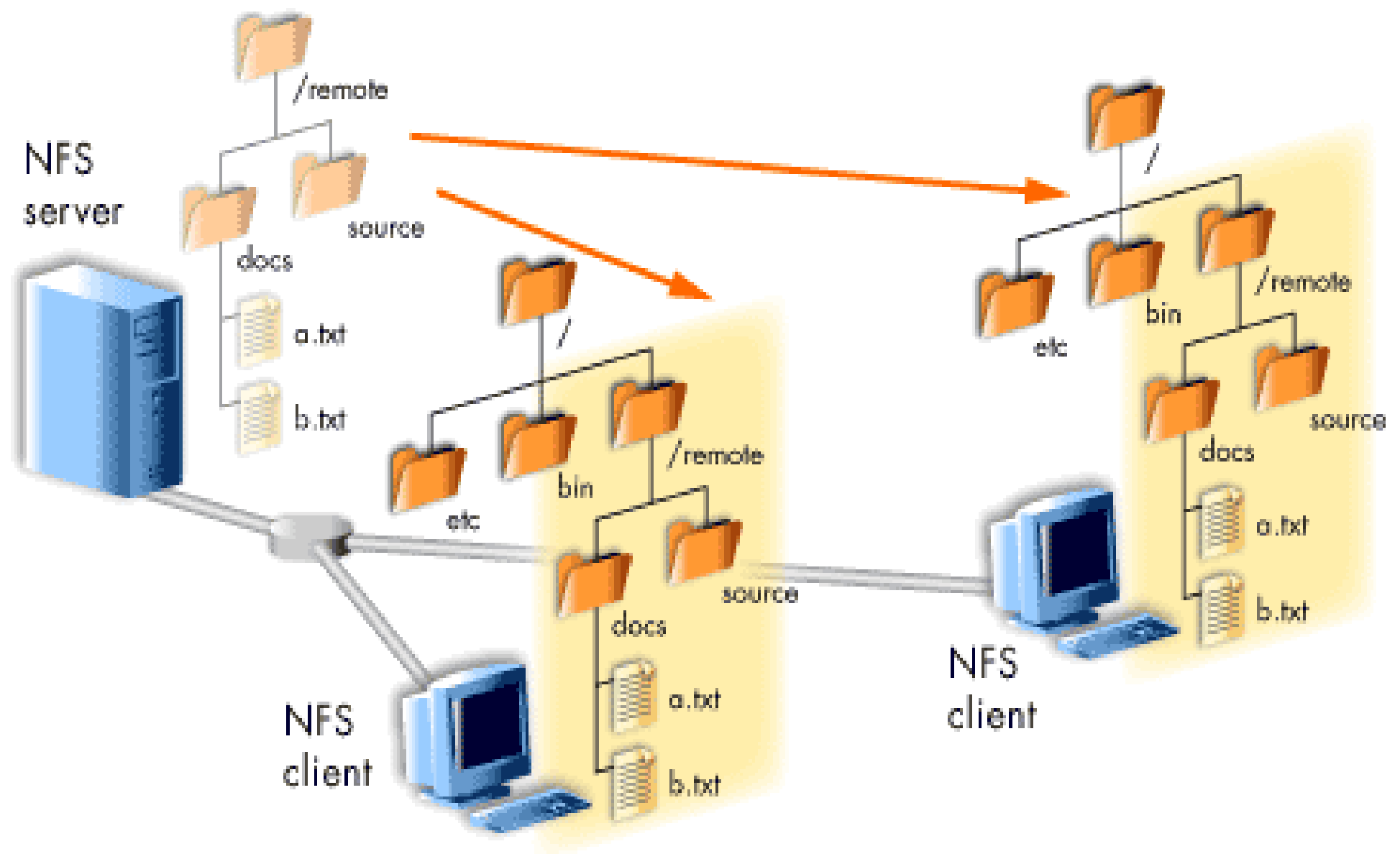
2-Network File System

ساختار فایل در
Unix

۲- Network File System یا NFS فایل سیستم هایی هستند که روی دیسکهای تحت شبکه کار می کنند شامل

موارد زیر می باشند

- NFS توسط شرکت : Sun Micro System. توسعه یافته
- SMB یا Server Message Block توسط : Microsoft SAMBA Services. توسعه یافته است
- Coda
- Gfs



The NFS server exports the `/remote` filesystem to two NFS clients.

NFS-SMB

مغز فایل در
Unix

NFS سیستم فایل شبکه اجازه می دهد تعداد زیادی کاربر یا سیستم دسته ای از فایلها را با استفاده از روش Client/Server. به اشتراک بگذارند NFS امکان به اشتراک گذاری تمامی سیستم فایلها گفته شده را می دهد

NFS حدود سال ۱۹۸۹ توسط Sun Microsystem ایجاد شد و در نسخه های مختلفی منتشر شد که ابتدا 2 و سپس نسخه ی 3 توسعه پیدا کردند و در حال حاضر نسخه ی 4 این سیستم فایل در حال استفاده است. البته روش های بهتری برای اشتراک فایل در شبکه وجود دارد که AFS" یا Andrew File System یکی از آن هاست اما استفاده و پیاده سازی این فایل سیستم ها به مراتب دشوار تر از NFS. می باشد

این سیستم در شبکه های یونیکسی NFS و در شبکه های مایکروسافت SMB نامیده می شود

با توجه به اینکه سرویس NFS دارای برخی ضعف های امنیتی است در صورت عدم پیکربندی مناسب آن سیستم های کامپیوتری در معرض تهدید بوده و مهاجمان با سوءاستفاده از حفره های امنیتی موجود قادر به دستیابی سیستم از راه دور و یا بصورت محلی خواهند بود.

NFS-SMB

ساختار فایل در
Unix

ساختار فایل به صورت زیر است :

(shared_directory IP or machin_name(option

shared_directory :

نام دایرکتوری از NFS Server است که برای دیگر ماشین های شبکه Share شده است.

IP or machin_name آدرس یا نام ماشین(های) کلاینتی است که مجاز به : mount. کردن دایرکتوری در سیستم محلیشان هستند

option گزینه هایی هستند مانند :rw. که بر شیوه استفاده در کلاینت از دایرکتوری اشتراکی اثر دارند

ro مخفف : Read Only که کلاینت های تنظیم شده با این گزینه، تنها دسترسی فقط خواندنی روی دایرکتوری Mount. شده دارند

rw مخفف : Read Write. که کلاینت های تنظیم شده با این گزینه، دسترسی خواندن و نوشتن بر روی دایرکتوری اشتراکی را دارند



NFS-SMB

مختصر فایل در
Unix

no_root_squash گزینه ای بسیار مهم در تنظیم یک دایرکتوری برای اشتراک گذاشتن، چونکه باعث کاهش امنیتی در سرور : NFS خواهد. اگر از این گزینه استفاده می کنید باید بدانید که کاربر root روی ماشین کلاینت، روی دایرکتوری اشتراک شده، دسترسی root مطابق با root. ماشین سرور را خواهد داشت و توصیه می شود که از این گزینه استفاده نکنید

root_squash نقطه مقابل : no_root_squash است. با استفاده از این گزینه، درخواست های آمده از uid=0 و gid=0 به کاربر anonymous که به nobody user یا nfsnobody user شناخته می شود، نگاشت خواهد شد. یعنی دسترسی کاربر root روی ماشین کلاینت بر روی دایرکتوری Share شده، معادل با دسترسی کاربر root روی ماشین سرور نباشد و باعث افزایش امنیت خواهد شد.

این گزینه باعث می شود که سرور تنها پس از نوشتن داده ها (اعمال تغییرات) کاملن تمام شد، به کلاینت پاسخ دهد. این : sync. گزینه، بعنوان پیشفرض است و اگر ننویسید، همین گزینه در نظر گرفته می شود

async نقطه مقابل : sync است و یعنی اینکه سرور قبل از تکمیل تغییرات بر روی Storage به در خواست های دیگر نیز پاسخ ، خواهد داد

4- راه اندازی مجدد سرویس به منظور اعمال تغییرات : `service nfs-kernel-server restart`

تنظیم سرویس گیرنده : 1- نصب بسته های `portmap` و `nfs-common` در کلاینت ها.

نکته : بسته مشترک میان دو ماشین سرور و کلاینت `portmap` است، که ارتباط میان سرور و کلاینت را برقرار می کند و در صورت غیر فعال (stop) بودن بر روی هر یک از ماشین ها باعث عدم دسترسی کلاینت مربوطه به پوشه اشتراکی خواهد شد.

2- ایجاد دایرکتوری لازم جهت اتصال (mount) به دایرکتوری اشتراکی توسط سرویس دهنده: `mkdir /mnt/nfs/home`

3- اتصال دایرکتوری ایجاد شده به دایرکتوری به اشتراک گذاری شده توسط سرویس دهنده به وسیله ی دستور زیر :

`mount -t nfs 192.168.110.100:/home /mnt/nfs/home`

لازم به ذکر است برای آنکه اجازه عبور ترافیک NFS داده شود دیواره آتش (iptables) را باید تنظیم کرد. در صورتی که سرویس دیواره آتش روی NFSserver فعال نباشد شاخه ی مورد نظر به سادگی توسط کاربر به سیستم متصل می شود اما در صورت فعال بودن سرویس دیواره ی آتش کاربر با خطای مواجه میشود. برای عملکرد صحیح سرویس `nfs` باید از فیلتر شدن پورتهای زیر توسط دیواره آتش جلوگیری نمود: `TCP:111 , UDP: 111 , TCP: 2049 , UDP:2049 , TCP: 892 , UDP: 892`

همچنین باید در فایل `etc/sysconfig/nfs/` مقدار `MOUNT_PORT=892` را فعال کرد و سرویس NFS را به صورت مجدد راه

اندازی نمود. [7]

NFS-SMB

ساختار فایل در
Unix

سیستم فایل SMB یا "Session Message Block" از طریق پروتوکل SMB و برنامه‌ی Samba در لینوکس قابل دسترسی است. SMB در حقیقت بری ویندوزهای 3x و 9x و NT منتشر شد و برای شبکه‌هایی که شامل ویندوز و لینوکس در کنار هم می‌شوند مناسب است. فایل سیستم "ncpfs" نیز برای پشتیبانی از پروتوکل "NCP" که در "NovellNetware" استفاده می‌شود قابل دسترسی است.



سیستم فایل دیگر Code نام دارد که یک فایل سیستم توزیع شده بر اساس AFS2 است که قابلیت‌های جدیدی مانند disconnect را برای کامپیوترهای سیار فراهم کرده و از نظر امنیتی و بازدهی نیز بسیار توسعه یافته است.



Other file system

سایر فایل در
Unix

سایر فایل سیستم‌ها (که ممکن است رابطه‌ای با لینوکس نداشته باشند و فقط جهت اطلاعات بیشتر نامبرده می‌شوند):

DTFS

QNX filesystem

EFS (Enhanced filesystem)

ODS (Open VMS file system)

ffS (BSD filesystem)

NSS (Novell storage services)

GPFS (general Parallel Filesystem)

NWFS (NetWare filesystem / 286)

HFS (HP-UX Hi performance filesystem)

NWFS (NetWare filesystem / 386)

LFS (linux log structured file system)

MFS (Last Macintosh File System)

Gfs(google file system)

RomFS



جمع بندی

این که از چه file system باید استفاده کرد کاملاً به شما بستگی دارد، مسائل بسیار زیادی می‌تواند در انتخاب یک فایل سیستم تاثیر داشته باشند، نسخه‌ی کرنلی که استفاده می‌کنید، حجم دیسک سختتان، اهمیت اطلاعات، نیاز به سرعت در دسترسی یا نیاز به بازیابی سریع‌تر و دلایلی بسیار. در اکثر مواقع این دلایل می‌توانند راه کارهایی کاملاً متضاد را در اختیار شما قرار دهند، برای نمونه در صورتی که برای بازیابی سریع اطلاعات اهمیت بیشتری قائل هستید دیگر به صرفه‌ی اقتصادی نباید توجه کنید، اگر نیاز به سرعت بالاتری برای دسترسی به اطلاعات احساس می‌کنید مجبور خواهید شد از بلاک‌هایی با سایز بزرگتر استفاده کنید که این نیز مقرون به صرفه نیست. اگر از امکانات شبکه استفاده می‌کنید مسلماً نیاز به یکی از فایل سیستم‌های شبکه خواهید داشت. توجه داشته باشید که انتخاب یک فایل سیستم مناسب می‌تواند برای شما بسیار حیاتی باشد، چرا که همان طور که در ابتدای مقاله اشاره کردم مهم‌ترین بخش فعالیت‌های روزانه‌ی هر سیستمی پرونده‌ها هستند، گزارش‌های روزانه، نامه‌های الکترونیکی و بسیاری فایل‌های دیگر همه و همه از طریق فایل سیستم مدیریت می‌شوند و انتخاب یک فایل سیستم مناسب بسیار حیاتی است.

البته گاهی نیز اجبارهایی در کار است، برای نمونه پارتیشن اصلی یا root را نمی‌توانید vfat انتخاب کنید! یا اگر از LILO به عنوان بوت لودرتان استفاده می‌کنید ممکن است نتوانید از XFS برای پارتیشن بوت استفاده کنید و یا ممکن است لازم باشد از ترکیب این فایل سیستم‌ها در کارتان استفاده کنید. برای نمونه از etx2 برای پارتیشن بوت، از دیگری برای / از فایل سیستمی سریع برای tmp/ از فایل سیستمی ایمین و قابل بازیابی برای دایرکتوری /home و ... که کاملاً به سیاست‌های شما و نوع کار شما وابسته است.

Jump up to: a b Russel Sandberg, David Goldberg, Steve Kleiman, Dan Walsh, Bob Lyon (1985). "Design and Implementation of the Sun Network Filesystem". USENIX.

http://en.wikipedia.org/wiki/Network_File_System

http://en.wikipedia.org/wiki/Network_File_System

<http://falearn.ir/>

http://docstore.mik.ua/orelly/networking_2ndEd/dns/index.htm

http://www.softpanorama.org/Internals/Unix_filesystems

<http://tldp.org/LDP/intro-linux/html/>