

به نام خدا

پروژه درس سیستم عامل | دکتر جلیلی

موضوع: بررسی filesystem Interface FreeBSD

محمد جواد علاءالدینی

زمستان ۱۴۰۰

فهرست مطالب

۳ مقدمه
۳ (File Operations) دستورات فایل
۳:Delete
۳:Read&Write
۳:Open & close
۳:Find
۴:Mkdir
۴ (File Locking) قفل فایل
۴ (Disk Organization) سازماندهی دیسک
۵:Hier
۵:Swap
۵:Disk
۵:Raid
۶ لیست دایرکتوری ها:
۹ (Types of File Systems) انواع فایل سیستم ها
۹:Ufs
۹:Zfs
۱۰ Directory Organization

مقدمه

FreeBSD یک سیستم عامل open source مانند یونیکس یا یک پلتفرم محبوب سرور است. این OS بسیار نزدیک به یک سیستم عامل نزدیک به یونیکس می باشد و از آن نشات می گیرد اما خب قاعدتا پیشرفته و کاربردی تر از آن خواهد بود. در این پروژه قصد داریم به بررسی رابط فایل سیستم در این سیستم عامل بپردازیم و قطعه کد های مربوطه را از درون کدهای FreeBSD استخراج کرده و آن ها را شرح دهیم.

سیستم عامل های مختلف در سیستم فایل خود متفاوت هستند و هرکدام سیستم فایل های خود را دارا می باشند. سیستم فایل های اصلی FreeBSD، سیستم فایل های Unix UFS می باشد که با عنوان به روز شده UFS2 منتشر شده است.

علاوه بر سیستم های فایل بومی خود، FreeBSD از بسیاری از سیستم های فایل دیگر پشتیبانی می کند تا داده های سایر سیستم عامل ها به صورت محلی قابل دسترسی باشند، مانند داده های ذخیره شده در دستگاه های ذخیره سازی USB متصل محلی، درایوهای فلش و دیسک های سخت.

دستورات فایل (File Operations)

Delete: با دستور rm توسط متوذهای خود در مود کرنل میتوان به حذف فایل ها و دایرکتوری ها پرداخت که در گیت هاب کد کامل آن آورده شده است.

Read&Write: خواندن و نوشتن فایل ها دستور دیگری است که در این سیستم عامل متد خاص خود را دارد. در گیت هاب با عنوان فایل read.c و write.c ذکر شده است که توسط یک تابع read و Write به خواندن بایت فایل ها میپردازد.

Open & close: باز و بسته کردن فایل ها از جمله دستورات فایل است که در اینجا با استفاده از متدهایی درون کلاس های open.c و close.c انجام میشود. ساز و کار باز کردن فایل ها در کد گیت هاب قابل مشاهده می باشد. Truncate: این دستور منجر به افزایش یا کاهش دادن طول یک فایل میشود و در گیت هاب کد کلاس آن موجود می باشد.

Find: ساختاری سلسله مراتبی دایرکتوری ها را طی میکند و به دنبال فایل مورد نظر میگردد؛ به صورت بازگشتی از بالا به پایین را طی میکند. کد کلاس آن در گیت هاب موجود است.

Mkdir: باعث ایجاد دایرکتوری به ترتیب میشود. در گیت هاب کلاس مورد نظر آن وارد شده است همچنین دستورات دیگری از جمله umask ماسک موردنظر برای ایجاد فایل را ایجاد میکند.

ls: این دستور در تمام سیستم عامل ها برای ما آشناست که با زدن آن در cmd هر سیستم عامل، لیستی از دایرکتوری هایی که در مکان آن قرار داریم برای ما ظاهر میشود. در گیت هاب کلاس ls.c قرار داده شده که کدهای فرایند ظاهر شدن ls را نشان میدهد.

قفل فایل (File Locking)

از انواع قفل گذاری ها در فایل سیستم عامل می توان به قفل گذاری اشتراکی و انحصاری اشاره کرد که هر کدام روی فایل ها اعمال میشود. قفل های انحصاری را هیچ کس نمیتواند بخواند زیرا در حال نوشته شدن هستند ولی در حالت اشتراکی چندین قفل مشترک باهم وجود دارد. پس قفل های اشتراکی برای چند فرایند و قفل های انحصاری تنها برای یک فرایند قابل استفاده هستند. این سیستم عامل به صورت اتوماتیک فایل های باز را قفل نمیکند. دو کلاس flock و fcntl در گیت هاب وجود دارد. همچنین کلاس sx نیز برای قفل های انحصاری اشتراکی کاربرد دارد

سازماندهی دیسک (Disk Organization)

کوچکترین واحد سازمانی ای که این سیستم برای یافتن فایل ها از آن استفاده میکند، نام فایل است. نام فایل case-sensitive می باشد. فایل ها در دایرکتوری ها ذخیره می شود. یک دایرکتوری می تواند فایلی وجود داشته نداشته باشد یا حاوی صدها فایل باشد. همچنین می تواند شامل دایرکتوری های دیگر باشد که به شما امکان ایجاد ساختار سلسه مراتبی دایرکتوری ها را می دهد. این امر سازماندهی داده های ما در این سیستم عامل را بسیار آسان می سازد. فایل ها و دایرکتوری ها با / رو به جلو به فایل های دیگر یا دایرکتوری های دیگر ارجاع داده می شوند. هر فایل سیستمی دقیقاً حاوی یک دایرکتوری در سطح بالاست که با آن دایرکتوری ریشه می گویند. دایرکتوری ریشه خود می تواند شامل دایرکتوری های دیگری نیز باشد. این روش جداسازی در سیستم عامل های مختلف متفاوت است.

در این سیستم عامل یک سیستم فایل به عنوان سیستم فایل ریشه تعیین می شود. دایرکتوری ریشه سیستم فایل ریشه به عنوان / نامیده می شود. سپس هر فایل سیستم دیگری در زیر فایل سیستم ریشه نصب می شود. مهم نیست که چه تعداد دیسک در سیستم FreeBSD خود دارید، به نظر می رسد هر دایرکتوری بخشی از همان دیسک است.

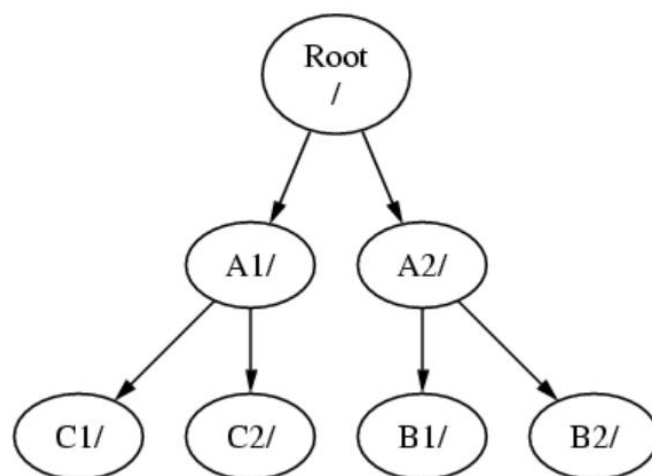


Figure 1 | نمونه از ساختار درختی یک دایرکتوری در FreeBSD

این روش معایب و مزایایی دارد که اکنون در این مقال نمیگنجد و گزارشات تکمیلی بعداً پیوست میگردد.

Hier: در کلاس hier گیت هاب سلسه مراتب این دایرکتوری ها و کاربرد آن مشخص شده است.

Swap: FreeBSD همچنین از فضای دیسک برای جابجایی فضای استفاده می کند. فضای Swap حافظه مجازی را برای FreeBSD فراهم می کند. این به رایانه شما اجازه می دهد تا طوری رفتار کند که گویی حافظه آن بسیار بیشتر از آنچه که واقعاً دارد دارد. وقتی حافظه FreeBSD تمام می شود، برخی از داده هایی را که در حال حاضر استفاده نمی شوند به فضای مبادله منتقل می کند و در صورت نیاز آن ها را دوباره به داخل منتقل می کند (چیز دیگری را خارج می کند).

Disk: عملیات های مختلف دیسک مثلاً باز کردن یک دیسک، اطلاعات یک دیسک، نوشتن و خواندن و... در در کلاس های مرتبط با دیسک انجام می پذیرد که در گیت هاب تحت کلاس های disk.c و diskinfo.c قرار داده شده است. این کلاس های شامل متود های انجام فعالیت های گفته شده می باشند.

Raid: در این سیستم عامل از روشی برای گروه بندی درایوهای فیزیکی مجزا در یک درایو بزرگتر پشتیبانی میشود. در این زمان سرور می تواند بین چندین درایو، خواندن و نوشتن را انجام دهد و کارایی در درایوهای رید شده بهتر است. رید ها توسط رید کنترلر، کنترل میشوند. رید کنترلر، لایه انتزاعی بین سیستم عامل و دیسک های فیزیکی است که گروه دیسک ها را به صورت واحدهای لاجیکال نمایش می دهد. در گیت هاب کلاس g_raid.c عملکردهای بالا را به تصویر می کشاند.

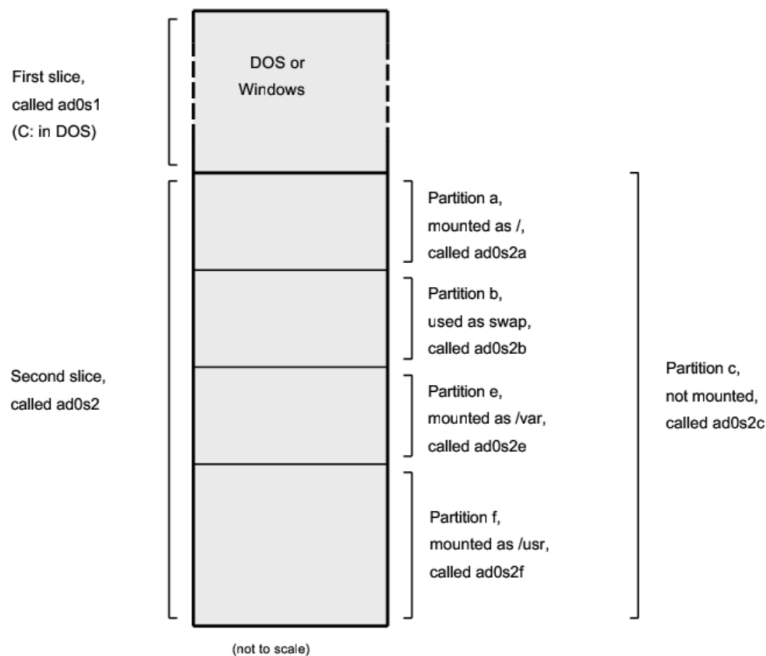


Figure ۲ مدل مفهومی از دیسک و پارتیشن بندی

لیست دایرکتوری ها:

لیست دایرکتوری ها Table 1

Directory	Description
/	Root directory of the file system.
/bin/	User utilities fundamental to both single-user and multi-user environments. Examples: cat; chflags; chmod; cp; csh; dd.....
/boot/	Programs and configuration files used during operating system bootstrap.
/cdrom/	default mount point for CD-ROM drives
/compat/	normally a link to <code>/usr/compat</code> . If not, then the <code>/usr/compat</code> comments apply

/dev/	Device nodes. Refer to intro(4) for details. Device special files managed by devfs(5)
/etc/	System configuration files and scripts.
/home/	is a link to <code>/usr/home</code>
/include/	new on GhostBSD without content
/lib/	critical system libraries needed for binaries in <code>/bin</code> and <code>/sbin</code>
/libdate/	new on GhostBSD without content
/libexec/	critical system utilities needed for binaries in <code>/bin</code> and <code>/sbin</code>
/man/	new on GhostBSD without content
/media/	contains subdirectories to be used as mount points for removable media such as CDs, USB drives, and floppy disks
/mnt/	Empty directory commonly used by system administrators as a temporary mount point.
/net/	automounted NFS shares; see auto_master(5)
/proc/	Process file system. Refer to procfs(5) , mount_procfs(8) for details.
/rescue/	Statically linked programs for emergency recovery as described in rescue(8) .
/root/	Home directory for the root account.

/sbin/	System programs and administration utilities fundamental to both single-user and multi-user environments.
/share/	new on GhostBSD without content
/tests/	new on GhostBSD without content
/tmp/	Temporary files which are usually not preserved across a system reboot. A memory-based file system is often mounted at <code>/tmp</code> . This can be automated using the tmpmfs-related variables of rc.conf(5) or with an entry in <code>/etc/fstab</code> ; refer to mdmfs(8) for details.
/usr/	The majority of user utilities and applications.
/var/	Multi-purpose log, temporary, transient, and spool files. A memory-based file system is sometimes mounted at <code>/var</code> . This can be automated using the varmfs-related variables in rc.conf(5) or with an entry in <code>/etc/fstab</code> ; refer to mdmfs(8) for details.
/www/	new on GhostBSD without content
file: /COPYRIGHT	opyright (c) 1992–2019 The FreeBSD Project.

انواع فایل سیستم ها (Types of File Systems)

سیستم عامل ها انواع مختلفی فایل سیستم چه برای استفاده خاص و چه برای استفاده عام دارند. از آنجا که freebsd سیستم عاملی یونیکس بیس است در نتیجه فایل سیستم های آن از سیستم عامل یونیکس نشات میگیرد. مانند سیستم فایل های ufs، zfs و proc

Ufs: یک استاندارد ذخیره سازی جهانی است که به منظور ارائه بیشترین سرعت خواندن و نوشتن اطلاعات به کار گرفته می شود. همچنین در مصرف انرژی بسیار به صرفه تر عمل میکند و علاوه بر سرعت زیاد مصرف کمی هم دارد. در گیت هاب فولدري از پیاده سازی ufs در سیستم عامل freebsd کامیت شده و قابل مشاهده است.

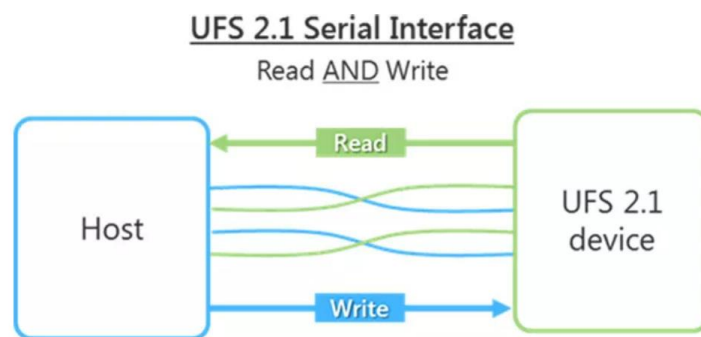


Figure ۳ شکل مفهومی یو اف اس

Zfs: علاوه بر نیرومندی، مقیاس پذیری و مدیریت آسان، دارای ویژگی ها و مزایایی می باشد که در هیچ یک از File System های بکار رفته در تجهیزاتی از قبیل IBM، EMC، HP و غیره ارائه نشده است. ZFS قابلیت های زیادی دارد که از آن یک File System مدرن ساخته است. برخی از این قابلیت ها عبارتند از: پشتیبانی از ظرفیت های بسیار بالا برای محیط ذخیره سازی، محافظت از اطلاعات در برابر آسیب دیدگی، پشتیبانی از SnapShot، سهمیه بندی دیسک، رمزنگاری اطلاعات، فشرده سازی اطلاعات (Compression)، جلوگیری از ذخیره داده های تکراری بر روی دیسک (Deduplication) و ...

ZFS تفاوت قابل توجهی با File System های ارائه شده ی قبل از خود دارد و حتی بیشتر از یک File System می باشد. ZFS قوانین و نقش های سنتی جداگانه مدیریت Volume و File System را ادغام کرده و یک File System یکپارچه با مزایای منحصر بفرد ارائه می دهد. File System قسمتی از ساختار دیسک می باشد. File System های قدیمی، در یک زمان فقط امکان پیاده سازی بر روی یک دیسک را دارند؛ لذا در صورت وجود دو دیسک، دو File System مجزا نیز ایجاد خواهد شد. که این مسئله با استفاده از RAID سخت افزاری و با ایجاد دیسک های منطقی، قابل پیاده سازی است. در گیت هاب کد پیاده سازی آن در سیستم عامل freebsd استخراج شده است.

Directory Organization

:Mtree