آموزش LPIC-1 (Exam 101,102)

مروری بر دوره

مدرک LPIC-1 اولین مدرک از مجموعه مدارک حرفه ای سطح بندی شده LPI لینوکس می باشد. این مدرک صلاحیت داوطلب در زمینه اداره کردن امور مرتبط با تعمیر و نگهداری از طریق خط فرمانی (command line)، نصب و پیکربندی یک کامپیوتر برخوردار از سیستم عامل لینوکس و نهایتا تنظیمات اولیه مرتبط با شبکه را تایید می کند.

دوره آموزش لینوکس در بین متخصصان فناوری اطلاعات (IT) از اهمیت و جایگاه ویژه ای برخوردار است. دلیل استقبال از دوره های Linux این است که لینوکس همچنان در حوزه رایانه جزو ابر قدرت ها و نام های شناخته این بازار به حساب می آید. طبق گزارشی که W3Techs برای سال ۲۰۱۹ میلادی منتشر کرده است بیش از ۵۰ درصد سهم بازار مربوط به وب سرور، به Linux اختصاص دارد.

در این دوره آموزشی سعی شده است در کنار مالتی مدیاهای آموزشی ناگفته های لینوکس را در قالب کتاب ارائه بدهیم، که به صورت کاملا عملی و سناریو محور می باشد. در واقع این کتاب مکمل مالتی مدیاهای آموزشی lpic-1 Exam 101,102 است.

این یک کتاب تئوری نیست بلکه در این کتاب صدها مثال کاربردی به همراه تصویر قرار داده شده است تا شما خودتان آنها را اجرا کنید و نحوه کار با دستورات لینوکسی را یاد بگیرید.

فهرست مطالب

- تغییر اولویت اجرای پروسه ها
- جستجو در فایلهای متنی با الگوهای متنی
- ویرایش فایلها با ادیتور vi

فصل چهارم : دستگاه ها، ساختار فایلها، استاندارد سلسله مراتبی فایل‌ها

- ایجاد پارتیشن و فایل های سیستمی
- کنترل سلامت فایل سیستم ها
- mount و unmount کردن فایل سیستم ها
- مدیریت ظرفیت فضای دیسک
- مدیریت مجوز فایل‌ها و مالک فایل‌ها
- ایجاد و تغییر لینک های سخت و symbolic لینک ها
- یافتن و قراردادن فایل‌ها در جایگاه صحیح

فصل پنجم : پوسته ها، اسکریپت ها و مدیریت داده ها

- دلخواه سازی و استفاده از محیط shell
- دلخواه سازی یا نوشتن اسکریپت های ساده
- مدیریت داده ها با SQL

فصل ششم : محیط کاربری و دسکتاپ

- نصب و پیکربندی محیط گرافیکی X11

- تنظیمات display manager
- امکانات دسترسی (معلولین)

فصل هفتم : وظایف مدیر سیستم

- مدیریت کاربران و گروه ها و فایل‌های مرتبط
- خودکارکردن وظایف سیستم
- محلی سازی و بین المللی سازی

فصل هشتم : سرویس‌های پایه ای سیستم

- مدیریت زمان
- مدیریت سیستم ثبت گزارش (log)
- مبانی انتقال نامه های الکترونیکی (Mail Transfer Agent)
- مدیریت چاپگرها

فصل نهم : مبانی شبکه

- مبانی پروتکل های اینترنت
- مبانی تنظیمات شبکه
- مبانی رفع اشکال شبکه
- استفاده از سیستم دامنه

فصل دهم : امنیت

- تنظیمات مقدماتی امنیت
- تنظیمات امنیتی Host
- مبانی رمز نگاری

فصل اول معماری سیستم

در این فصل به بررسی قسمت های مختلف سیستم عامل لینوکس می پردازیم .

فرض را بر این می گیریم که شما به یک کامپیوتر که بر روی آن لینوکس نصب شده است دسترسی دارید و در Shell یا Terminal آن دستوراتی را که لازم باشد را می توانید وارد کنید و خروجی ها رو مشاهده کنید.

مواردی که مورد بررسی قرار می دهیم :

- تشخیص و پیکربندی تنظیمات سخت افزاری
- بالا آمدن (Boot) سیستم
- تغییر Runlevel ها و خاموش یا Reboot کردن سیستم

معماری سیستم

همان طور که می دانید همه سیستم عامل ها بر روی سخت افزار اجرا می شوند و این سخت افزار بر روند اجرای سیستم عامل تأثیر می گذارد. مثلا سخت افزار می تواند سریع یا کند، قابل اتکا یا دچار مشکل باشد .

پس تمام کامپیوترها به همراه مجموعه ای از قطعات سخت افزاری روانه ی بازار می شوند که شامل مواردی مانند پردازنده ی مرکزی ، RAM و هارد دیسک هستند . در قلب بسیاری از این قطعات سخت افزاری Firmware قرار دارد که ابزارهای پیکربندی رو فراهم می آورد و روند بالا آمدن سیستم عامل را هم آغاز می کند. با مطالعه ی دفترچه ی راهنمای بسیاری از قطعات سخت افزاری می توانید در مورد Firmware آنها اطلاعات زیادی کسب کنید، اما به یاد داشته باشید که همین که سیستم عامل لینوکس بالا آمد، مدیریت دستگاه های سخت افزاری از طریق ابزارهایی خواهد بود که سیستم عامل لینوکس فراهم آورده است . قسمت های کلیدی که توسط Firmware و بعد از اینکه سیستم عامل بالا آمد، توسط لینوکس مدیریت می شوند عبارتند از: وقفه ها ، آدرس های ورودی و خروجی ، دسترسی مستقیم به حافظه و واسط های ATA هارد دیسک ها که خود بر دو نوع SATA و PATA هستند که اخیراً نوع SATA آن رایج تر شده است.

Firmware

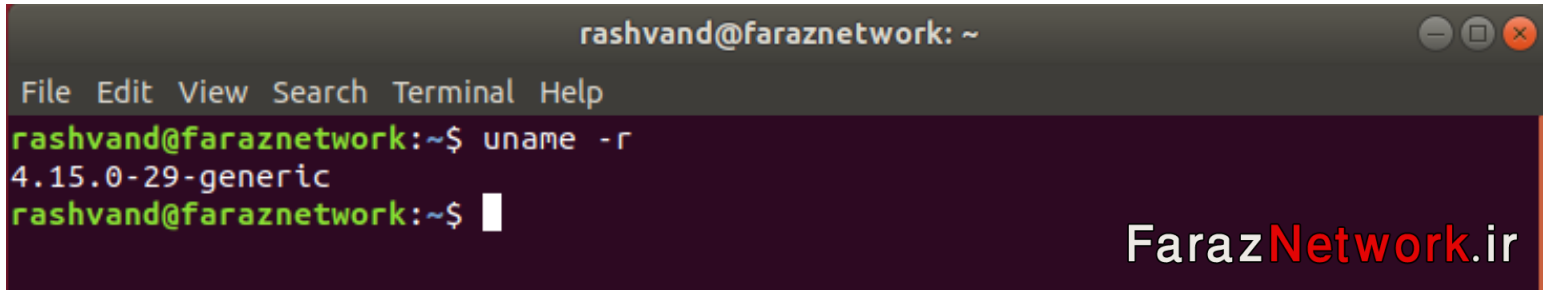
مهمترین Firmware همانی است که بر روی مادربرد سیستم نصب شده است، که نقش راه اندازی قطعات و کنترلرهای سخت افزاری مادربرد کامپیوتر و کنترل روند بالا آمدن سیستم رو بر عهده دارد. تا اواخر سال 2010 اکثر سیستم ها مبتنی بر BIOS بودند که مخفف عبارت Basic Input/Output System است .

از اوایل سال ۲۰۱۱ سیستم های جدید، مبتنی بر Firmware ی هستند که EFI نام دارد که مخفف Extended Firmware Interface است. نسخه ی تازه تر EFI که EFI 0.2 یا همان UEFI نام دارد، به کمک دو شرکت Microsoft و Intel توسعه یافته است که امکانات بسیار بیشتری را در مقایسه با نسخه های قبلی در اختیار مسئول سیستم قرار می دهد. بنابراین استاندارد موجود برای سیستم های کامپیوتری جدید UEFI است . البته به خاطر حفظ سازگاری با نسخه های قبلی سیستم های کامپیوتری بیشتر نسخه های UEFI امکان کار کردن در حالت BIOS را هم دارند. در صورتی که قطعه ای از قطعات سخت افزاری قدیمی باشد، باز هم امکان کار با آن وجود دارد . در ادامه ابتدا به پیکربندی سخت افزار می پردازیم و بعد هم به دقت روند بالا آمدن سیستم رو بررسی خواهیم کرد.

بررسی قطعات USB

توزیع های جدید لینوکس به همراه راه اندازهای USB ارائه شده اند و نیازی نیست که برای اتصال و راه اندازی قطعات USB معمولی کار خاصی انجام بدهید. بنابراین اگر مثلا یک کیبورد USB رو به کامپیوترتان که بر روی آن لینوکس نصب شده است وصل کنید، لینوکس آن را خواهد شناخت.

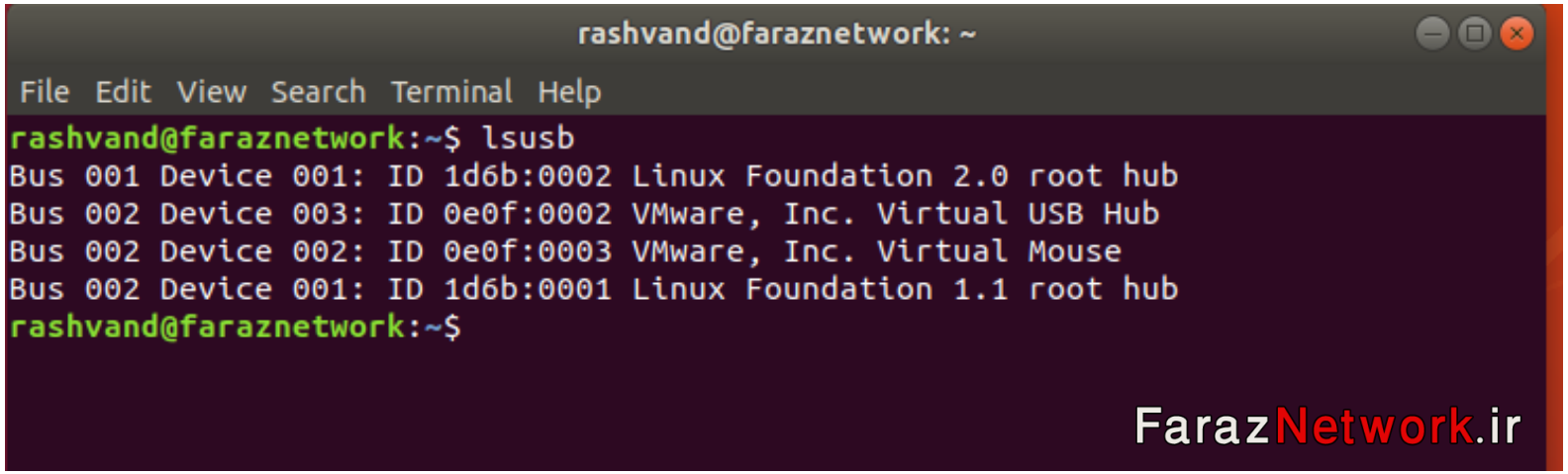
نکته : به خاطر داشته باشید که برای استفاده از USB 0.3 باید نسخه ی کرنل لینوکسی که از آن استفاده می کنید 2.6,31 یا بالاتر باشد. برای دیدن نسخه ی کرنل لینوکسی که دارید دستور uname -r را بزنید.



در خط فرمان لینوکس دستور زیر رو وارد کنید و سپس کلید Enter رو فشار بدهید :

```
$ lsusb
```

خروجی اون به مشابه خروجی زیر خواهد بود :



با توجه به خروجی این دستور متوجه میشویم که 4 باس USB تشخیص داده شده است (بزرگترین شماره در ستون دوم 003 است) در مثال ما هیچ دستگاهی به پورت های USB وصل نیست. اگر دستگاهی به پورت های USB سیستمی که شما بر روی آن lsusb را اجرا می کنید متصل باشد، در خروجی دستور lsusb اطلاعات آنها را مشاهده خواهید کرد.

معماری سیستم

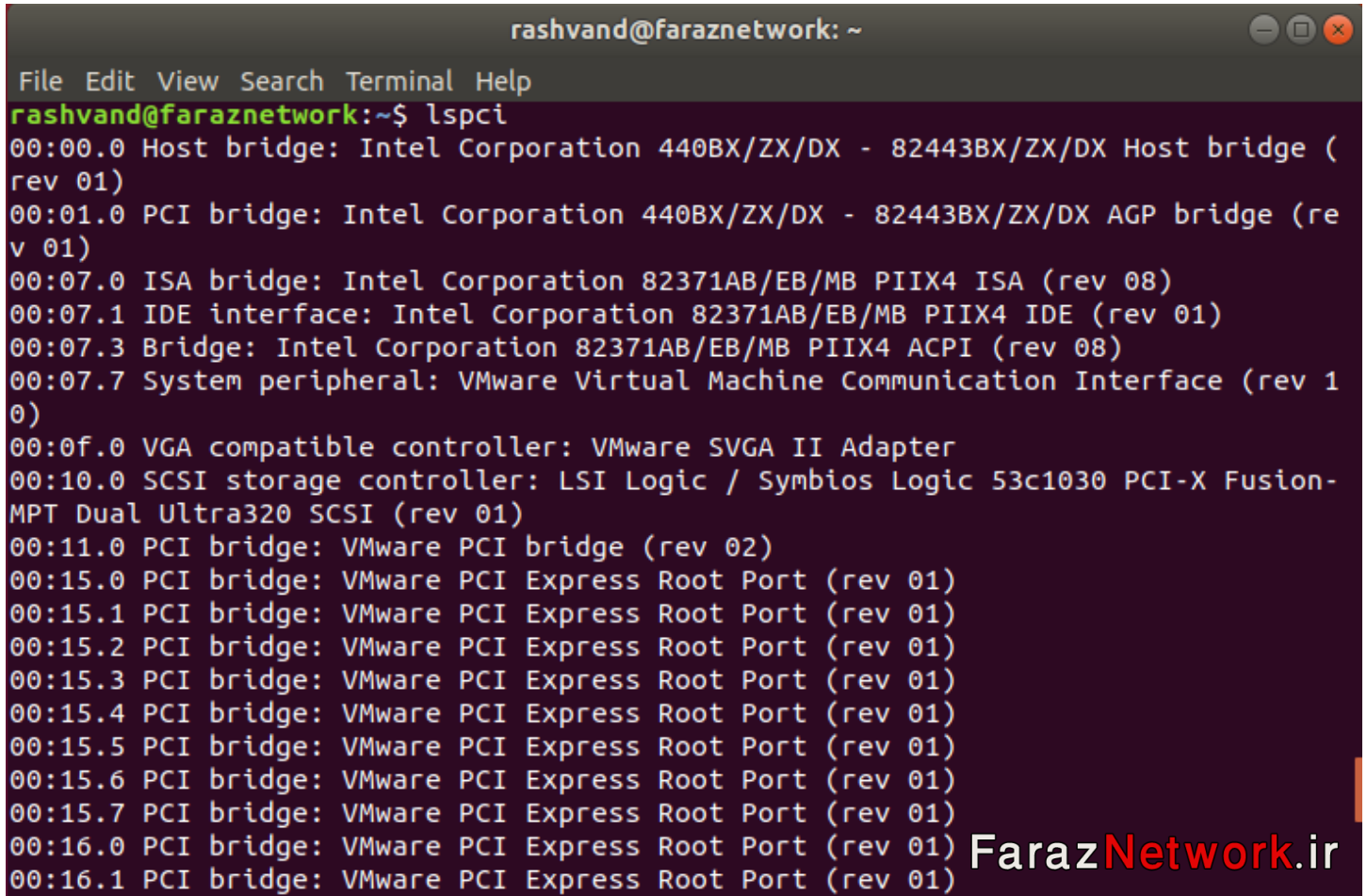
توصیه می کنیم که حتماً به صفحه ی راهنمای این دستور مراجعه کنید. برای این کار دستور man lsusb رو بزنید. این دستور آرگومان های زیادی را به شما نمایش می دهد که مهمترین آنها را در جدول زیر مشاهده می کنید:

آرگومان	توضیح
-v	در خروجی دستور توضیحات بیشتری رو به ما نمایش میدهد.
-s [[bus]:]devnum]	در خروجی تنها اطلا عات مربوط به bus و devnum داده شده رو چاپ میکند.
-d [vendor]:[product]	به کمک این آرگومان می توانید اطلاعات مربوط به محصول خاصی از تولید کننده خاصی رو ببینید.
-t	اطلاعات را در خروجی به صورت درختی نمایش می دهد، به این ترتیب راحتر می توان تشخیص داد که چه دستگاهی به چه کنترلری متصل شده است.

دستور lspci

علاوه بر دستگاه های USB ،قطعاتی داریم که بر روی باس های PCI کامپیوتر قرار دارند. جهت مشاهده ی پیکربندی دستگاه های PCI دستور زیر را بزنید:

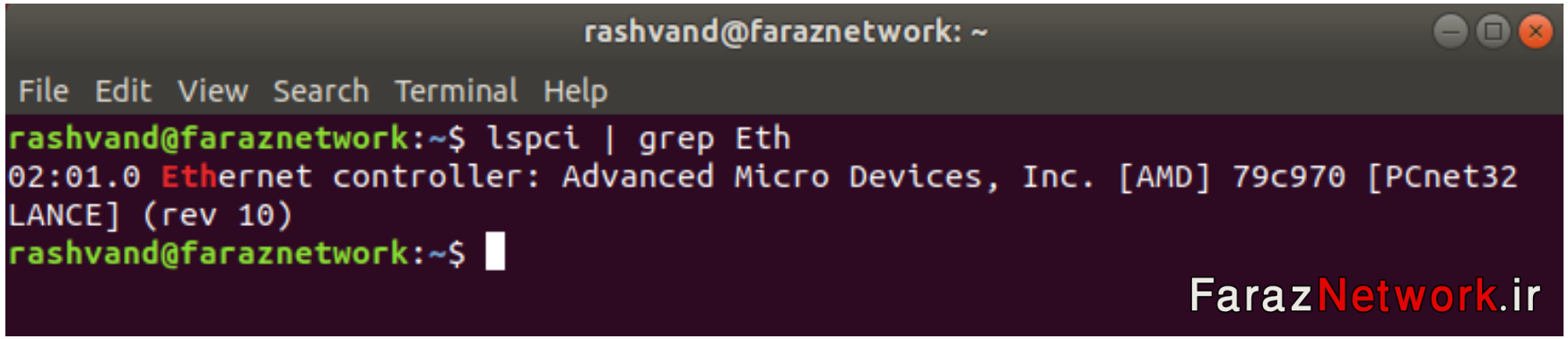
$ lspci



مثلا فرض کنید که می خواهیم پیکربندی PCI کارت شبکه را مشاهده کنیم، دستور زیر را وارد می کنیم :

```
$ lspci | grep Eth
```

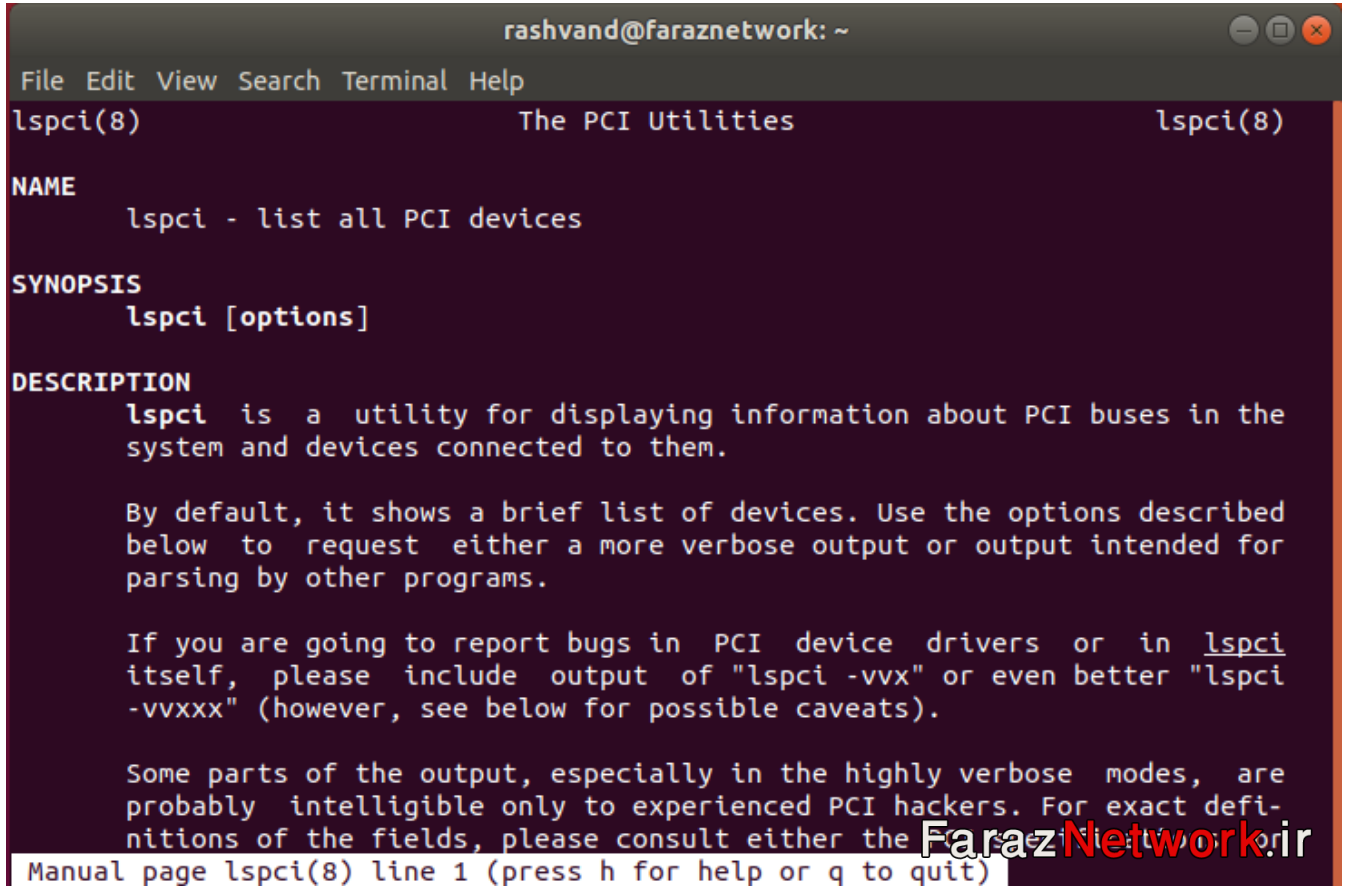
که بر روی سیستم ما خروجی زیر را تولید می کند:



پس ما یک پورت شبکه بر روی سیستم داریم. آدرس 02:01.0 برای این کارت شبکه به این معناست که این کارت شبکه بر رویBUS با شماره ی 02 ،اولین قطعه (01) سخت افزاری است (معمولا اولین قطعه سرعت بالقوه ی بالاتری دارد) و0. هم شماره ی Function آن را نمایش می دهد .پس 02:01.0 به صورت Bus: Device Function اطلاعات پیکربندی سخت افزاری را نمایش می دهد.

یکی از نکات مهمی که باید به خاطر داشته باشید این است که تا زمانی که محل اتصال قطعات سخت افزاری را در داخل کامپیوتر جا به جا نکرده باشید، این اطلاعات ثابت خواهند بود. به خصوص این مسئله برای کارت های شبکه اهمیت خاصی دارد چون ممکن است با ایجاد تغییرات نرم افزاری نام آنها عوض شود، مثلا eth1 تبدیل شود به eth4 درحالی که اطلاعات پیکربندی سخت افزاری آن تا زمانی که جابه جایی سخت افزاری صورت نداده باشید، ثابت خواهد بود (معمولا تغییرات نرم افزاری بسیار بیشتر از تغییرات سخت افزاری است)

توصیه می کنیم که حتماً با اجرای دستور man lspci به صفحه ی راهنمای این دستور مراجعه نمایید

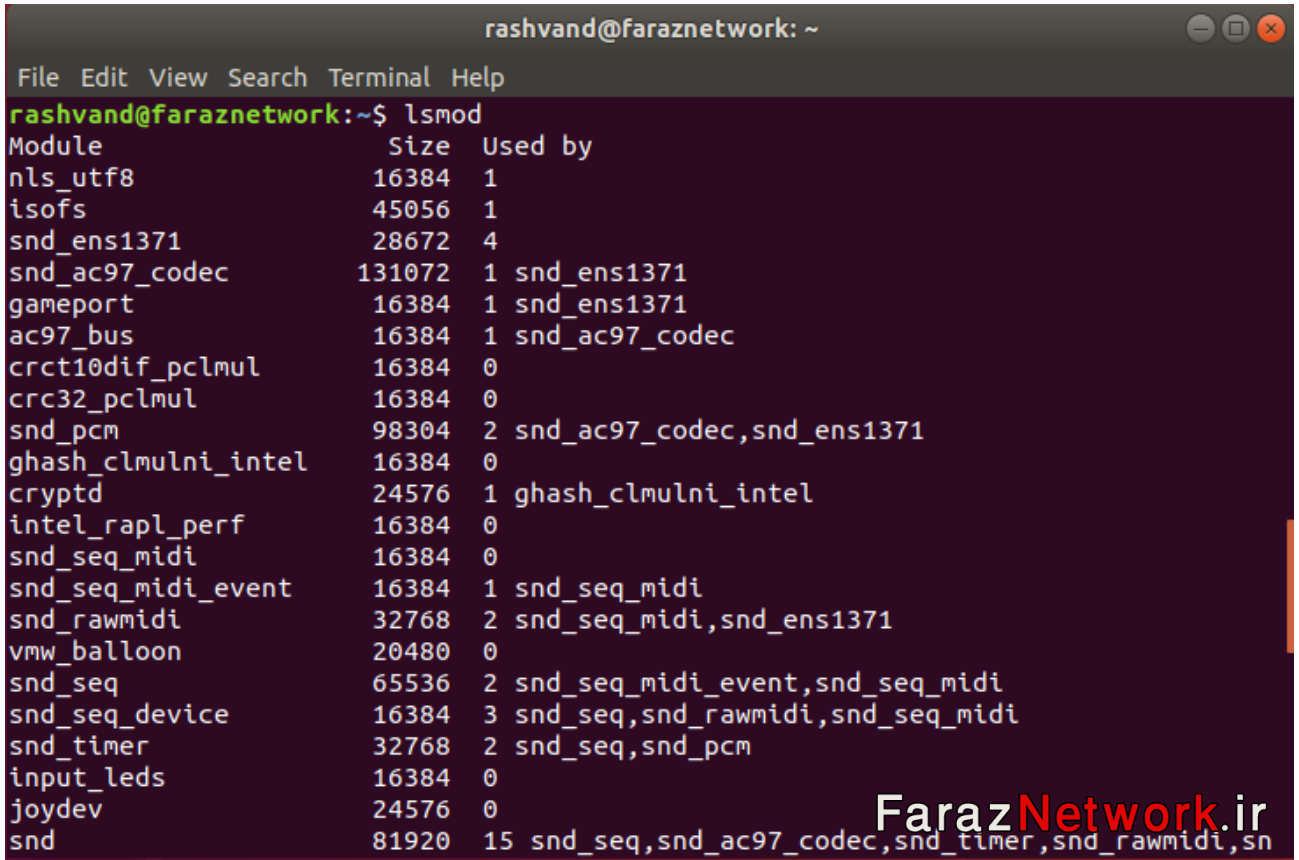


آشنایی با Kernel Modules

قطعات سخت افزاری در لینوکس توسط راه اندازهای کرنل (هسته ی سیستم عامل لینوکس) کنترل می شوند. بسیاری از راه اندازهای کرنل به فرم ماژول هستند که فایل های مستقلی هستند که معمولاً در مسیر /lib/modules قرار دارند که می توانند جهت فراهم آوردن دسترسی به قطعات سخت افزاری مرتبط با خودشان بارگذاری شوند، یا جهت غیرفعال کردن چنان دسترسی، Unload شوند. لینوکس در هنگام بالا آمدن ماژول هایی را که نیاز دارد، را بارگذاری می کند. فقط در صورتیکه نیاز به بارگذاری ماژول های اضافی که خودتان نیاز دارید، ممکن است که لازم باشد که توانایی Load یا Unload کردن ماژول ها را داشته باشید. دستور زیر را وارد کنید :

$ lsmod

خروجی آن به این صورت خواهد بود :



ستون اول نام تمام Module هایی را که بارگذاری شده اند نمایش می دهد. برای کسب اطلاعات بیشتر در مورد Module خاصی می توانید دستور modinfo و بعد نام آن ماژول را وارد کنید. ستون دوم نمایش دهنده ی بخشهایی است که از آن ماژول استفاده می کنند.به خاطر داشته باشید، که خروجی دستور lspci ممکن است بر روی برخی سیستم ها بسیار طولانی شود، به همین دلیل توصیه می کنیم که آن را به صورت lsmod | less اجرا کنید.

نکته مهم: ممکن است شما دو کامپیوتر داشته باشید که سخت افزارهای کاملاً یکسانی دارند ولی خروجی دستور lsmod بر روی آنها متفاوت است . یکی از دلایلی که میتواند باعث این مسئله شود این است که امکان Compile کردن و چسباندن Module به خود هسته ی لینوکس وجود دارد که در این صورت دیگر نام آن در خروجی lsmod نخواهد آمد، زیرا که آن Module دیگر بخشی از هسته ی سیستم عامل محسوب می شود و از همان ابتدای کار که هسته بارگذاری می شود در حافظه ی اصلی حضور خواهد داشت .برای بارگذاری Moduleی که نیاز دارید ولی قبلاً بارگذاری نشده است ، میتوانید از دو دستور insmod و modprobe استفاده کنید. به عنوان مثال:

```
$ modprobe floppy
```

همان کاری را انجام می دهد که فرمان

```
$ insmod /lib/modules/2.6.26/kernel/drivers/block/floppy.k
```

انجام خواهد داد. معمولاًروش اول توصیه می شود.

برای Unload کردن یک Module که قبلاً بارگذاری شده است می توانید از دستور modprobe -r یا rmmode استفاده کنید.

نکته : جهت اطلاعات بیشتر صفحات man دستورات گفته شده در بالا را مطالعه کنید. مثلاً دستور man modprobe را وارد کنید و بعد آن را مطالعه کنید.

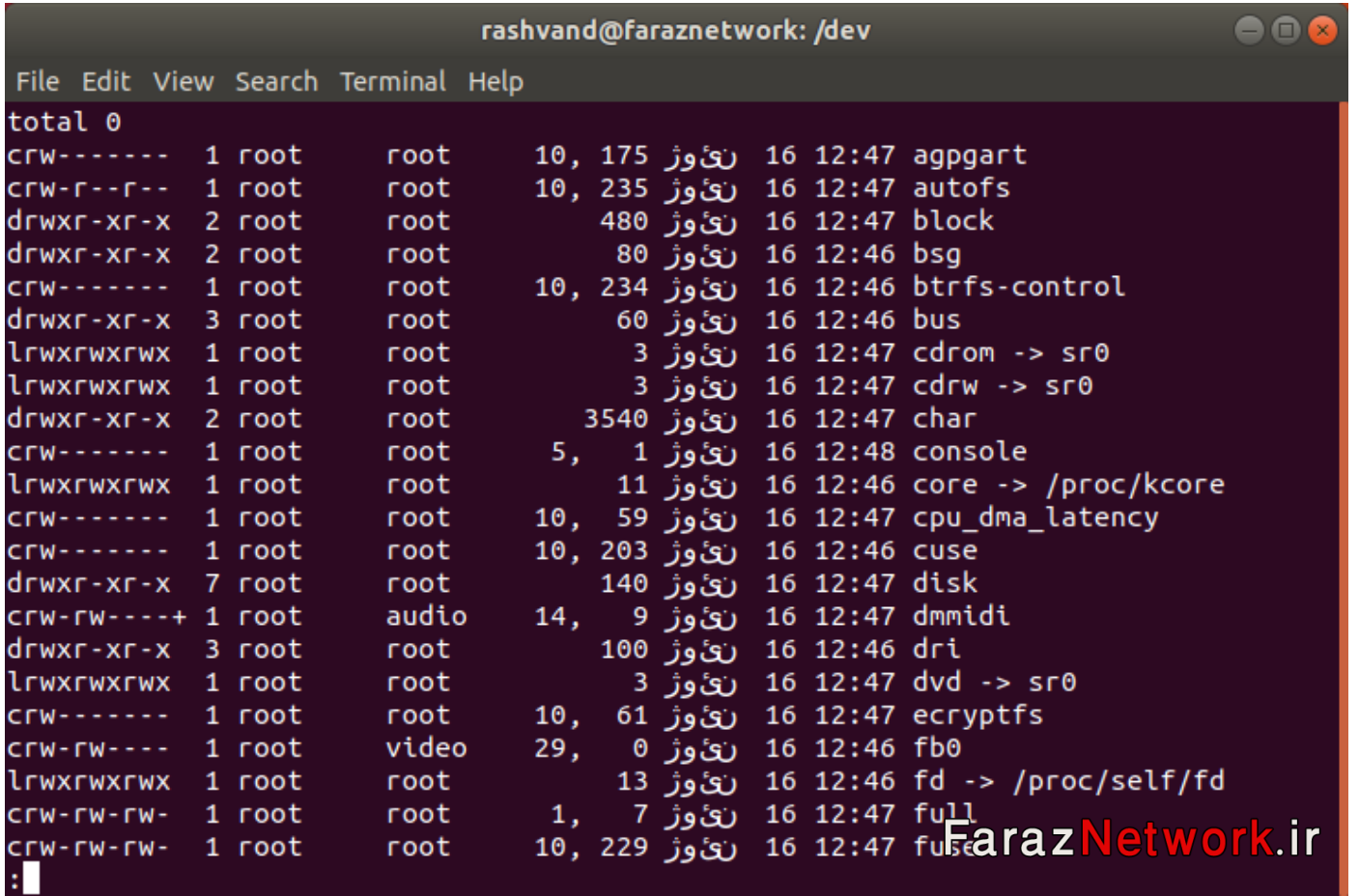
بررسی دایرکتوری /dev

این دایرکتوری شامل فایل های ویژه جهت نمایش دستگاه های متصل شده به سیستم است .

دستورات زیر را اجرا کنید:

$ cd /dev

$ ls -l | less



برای رفع ابهام، در این قسمت تنها یکبار به صورت بسیار جزئی و ساده نحوه ی نوشتن دستور بالا را توضیح می دهیم:
توضیح دستور بالا: بنویسید ls سپس یک فاصله قرار دهید، بعد علامت منها را بزنید و بعد از آن، بدون فا صله، ال کوچک انگلیسی را تایپ کنید. حالا یک فاصله قرار دهید و کلید شیفت را نگه دارید و کلید \ را بر روی کیبورد خود فشار دهید تا کاراکتر | که به آن Pipe می گویند نوشته شود. حالا یک فاصله قرار دهید و بنویسید .less بعد از اجرای دستور بالا، با فشار دادن فلش های بالا و پایین می توانید در متن خروجی، در صورتی که بیش از یک صفحه باشد، حرکت کنید. برای خروج کلید q را فشار دهید.

با دقت در خروجی دستورات بالا متوجه می شویم که در زیر دایرکتوری dev/ دستگاه هایی (Devices) که به سیستم متصل هستند، مثل ماوس، کیبورد، هارد دیسک و ... قرار دارند.
در خروجی دستور بالا (ls –l | less) سمت راست ترین فیلد اسم فایل، در کنارش تاریخ و ساعت و مالک فایل و سپس مجوزهای دسترسی و استفاده از فایل است . در لینوکس تقریباً همه چیز یک نمایش به صورت فایل دارد، بنابراین مثلاً فلش درایو شما می تواند در آدرس dev/sdb/ نمود فایلی داشته باشد.

تشخیص نوع هارد دیسک و دستگاه های کامپیوتر از روی نام فایل در زیر dev/

در زیر دایرکتوری dev/ فایل هایی وجود دارند که نماینده ی دستگاه های سخت افزاری هستند. فایل های مربوط به هارددیسک های IDE معمولاًبه صورت dev/hdX/ است که در آن X یک حرف است ، مثلاً hda و در صورتی که این دیسک پارتیشن بندی شده باشد، مواردی همچون dev/hda1/ و dev/hda2/ نیز خواهیم داشت (که مثلاً دو عدد پارتیشن بر روی این هارددیسک پیکربندی شده است). در صورتی که هارددیسک از نوع SATA باشد به صورت نشان داد ه می شود. درایوهای فلش نیز به صورت dev/sdX/ نمایش داده می شوند. CDROM و DVDROM نیز به صورت dev/scdX/ و dev/srX/ نمود پیدا می کنند.

دسته بندی دستگاه ها (Devices) از یک دیدگاه می توان دستگاه های کامپیوتری را به دو دسته تقسیم کرد :

۱. دستگاه های بلوکی (Block Devices) : تبادل اطلاعات میان چنین دستگاهی و سایر بخشهای کامپیوتر به صورت بلوک به بلوک است هارد دیسک یک نمونه از این نوع دستگاه ها است. یا مثلاً اگر بخواهید از CD ROM یک بایت بخوانید مجبور هستید که یک Block بخوانید و سپس از آن Block خوانده شده، بایت مورد نظر شما برای شما برگردانده می شود؛ دلیل این است که اطلاعات به صورت بلوک در CDROM ذخیره شده است . خواندن به صورت بلوکی از این دستگاه ها به دلیل نیاز به کارآیی بالاست ، زیرا که خواندن به صورت بیت به بیت سرعت انتقال داده ها را پایین می آورد.

۲. دستگاه های کاراکتری (Character Devices) : این نوع از دستگاه ها اطلاعات را به صورت حرف به حرف برمی گردانند. به عنوان مثال کیبورد، مودم و پرینتر در این دسته قرار می گیرند.

در سمت چپ خروجی دستور (ls –l | less) در زیر دایرکتوری /dev مجوزهای دسترسی نو شته شده است . اولین حرف این مجوزها، معنای خاصی دارد.

اولین حرف	نمایش دهنده
d	دایرکتوری
خط تیره (-)	File
b	Block Device
p	Character Device
|	Pipe
s	Socket

پس b برای دستگاه های بلوکی و c برای دستگاه های کاراکتری است .

نمایش محتوای فایل

با فرمان cat میتوان محتوای فایل را نشان داد. به عنوان مثال برای اینکه محتویات هارد دیسک را نشان دهیم، فرمان زیر را اجرا می کنیم :

```
$ cat /dev/sda
```

از آنجا که خروجی ممکن است طولانی باشد و نمایش آن زمان بر شود، بعد از اجرای دستور جهت خاتمه ی کار می توانید از ترکیب کلیدهای Control+C استفاده کنید.

روش دادن اسم دوّم به یک Device

یکی از روشهای انجام این کار استفاده از فرمان ln است :

```
$ ln /dev/fd0 a:
$ ls a: fd0
4349                a:
4349                fd
```

نکته ی کاربردی: یکی از ویژگی های خوب دایرکتوری dev این است که می توانید بفهمید آخرین باری که یک Device کار کرده چه زمانی بوده است . به این منظور کافی است به تاریخ ایجاد فایل ها در خروجی دستور ls –alh نگاه کنید.

حالا می خواهیم ببینیم چند تا Printer داریم و پورت های آنها را ببینیم :

```
$ ls –l /dev/lp*
crw-rw----1 root lp 6,0 aug 18 2006 jan fd0
```

در اینجا ما یک پرینتر داریم. ممکن است در سیستم شما پرینتر وصل باشد یا نباشد.

تعریف یک دستگاه جدید با دستور mknod

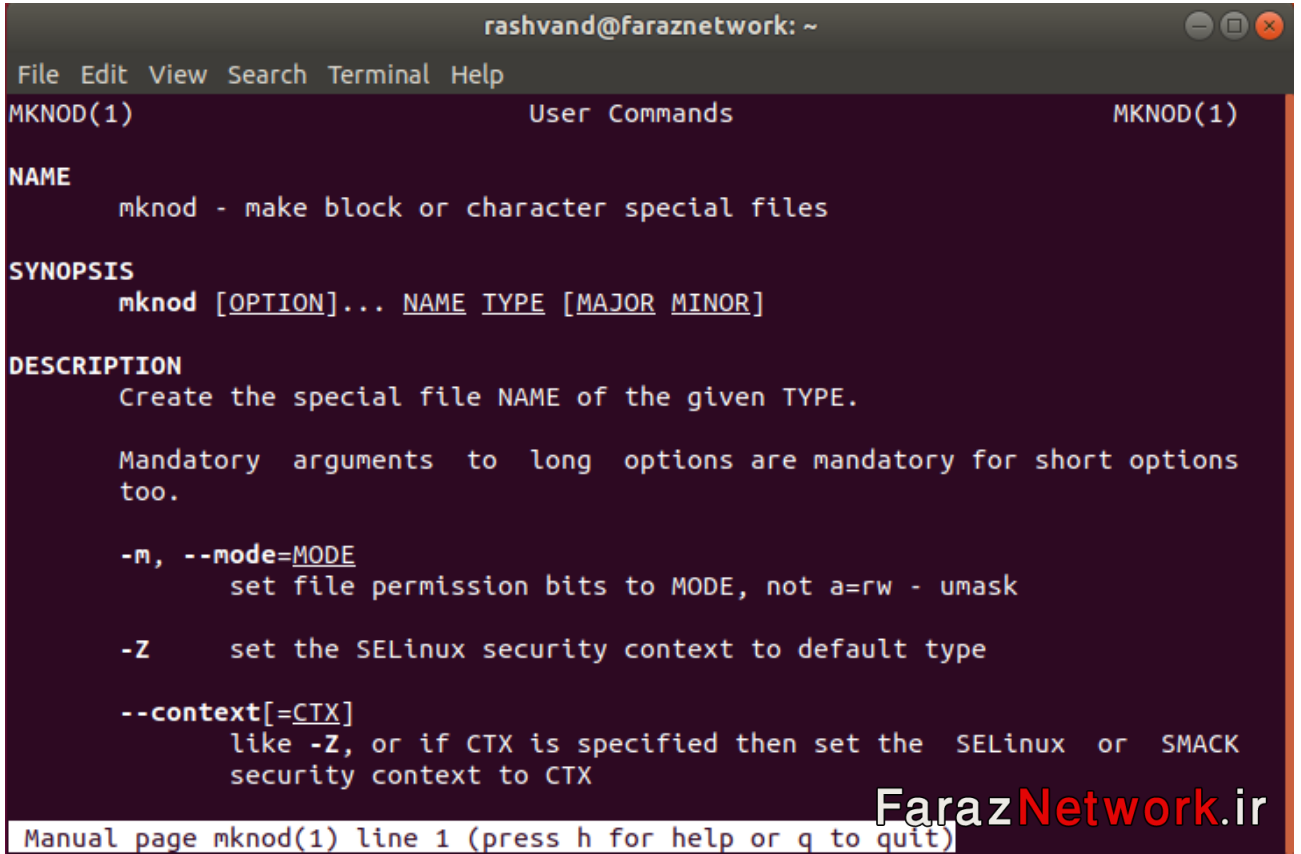
```
$ cd /dev
$ mknodlp1 c 6 1
name: lp1ß
type: c
major: 6
minor: 1
```

با دستور بالا یک پرینتر دیگر هم تعریف کردیم. البته توجه کنید که متناظر با پرینتر با نام lp1 دستگاه پرینتر وجود ندارد، بنابراین با ارسال فایل به این Device چیزی چاپ نخواهد شد. عدد ۶ عدد major به معنی اصلی است که هسته ی لینوکس (Kernel) با استفاده از آن تعیین می کند که چه Device Driver ی را انتخاب کند و با آن کار کند. عدد دوّم را minor می گوییم و با استفاده از آن معلوم می شود که یک Device چه مدلی دارد و همچنین کدام یک از پرینترهای متصل به کامپیوتر است . type: c نشان دهنده ی نوع Device است که در اینجا با نوع دستگاه کاراکتری سر و کار داریم پس برخی فایل ها در زیر دایرکتوری /dev نشان دهنده ی دستگاه های Character ی یا Block ی هستند و برخلاف فا یل های معمولی قد و حجم ندار ند، بلکه major و minor دارند که major شماره Device است و minor تعداد دستگاه با آن نوع را مشتتخص می کند. مثلاً وقتی چند تا هارد دیسک به یک کامپیوتر متصل هستند که از یک نوع Driver استفاده می کنند برای تشخیص آنها از یکدیگر کافی است که به عدد minor مراجعه شود. به این ترتیب با این دو عدد Device ها از هم تفکیک و شناخته می شوند.

یادتان باشد که فقط Root اجازه ساخت فایل های Device را دارد، و در واقع به طورمعمول این فایل ها را ما نمی سازیم و معمولاً این خود سیستم عامل است که بعد از متصل شدن دستگاه به کامپیوتر آن را تشخیص می دهد و نمود فایلی آن را در زیر دایرکتوری /dev می آورد. بنابراین فایل های زیر دایرکتوری dev پویا هستند و خود سیستم Device هایش را شناسایی می کند.

پس در این قسمت یاد گرفتیم که فرمانی که امکان ساخت Device File را میدهد mknod است و البته فقط کاربر Root امکان استفاده از آن را دارد. برای مشاهده ی صفحه ی راهنمای دستور mknod، دستور زیر را اجرا کنید :

```
$ man mknod
```



حال به عنوان Root دستور زیر را اجرا کنید :

```
$ cd /dev
$ su
# mknod  faraz b 250 44
```

b یعنی Block Device و شماره ی اول major و شماره ی بعدی minor است . البته ما اجازه ساخت نداریم چون در کامپیوتر ما برای هسته ی لینوکس دستگاهی با شماره ی major ۲۵۰ وجود ندارد، پس پیغام No Such Device را مشاهده می کنیم. پس به جای اینکه به طورتصادفی یک عدد major بنویسیم به خروجی دستور زیر مراجعه می کنیم و متناسب با دستگاه مورد نظرمان شماره ی major را انتخاب می کنیم :

```
$ cat/proc/devices
```

به جز c، d، b و – ما در سمت چپ دایرکتوری dev حرف های دیگری هم داریم مثل p به فرمان های زیر توجه کنید:

```
$ mknod faraz p
$ ls –l
total 8
faraz تاریخ prw-r--r--1 n. faraz lpi1 0
```

یک Pipe File ساختیم. Pipe File ها که معمولاً ابتدا و انتها ندارند و اطلاعات پشت سر هم نمی آید. به طورمعمول Pipe File ها خیلی کاربرد دارند و البته اطلاعات شان هم تمامی ندارد. از جلمه دستگاه هایی که به صورت Pipe File نمایش داده می شوند رطوب سنج و رادار هستند.

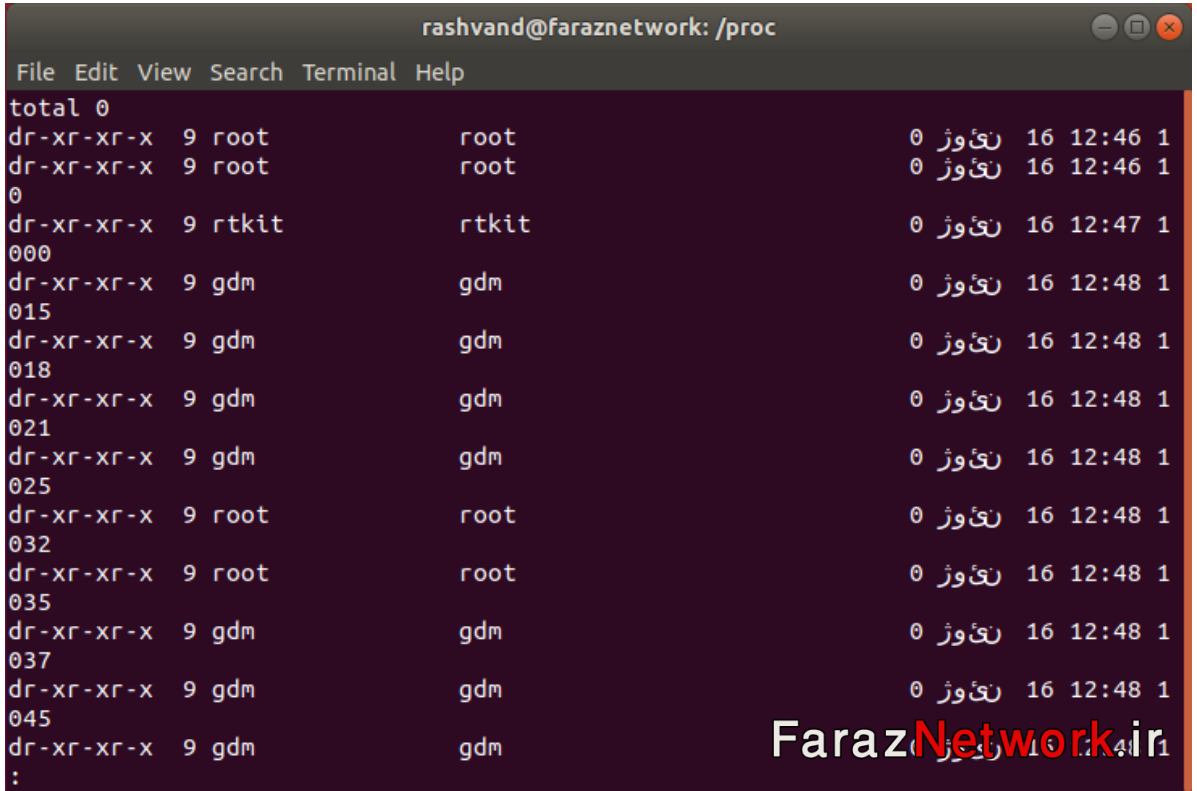
علاوه بر فرمان mknod در لینوکس یک فرمان دیگر به نام MAKEDEV (با حروف بزرگ) وجود دارد و به منظور ایجاد فایل دستگاه ها در زیر دایرکتوری dev به کار می رود (جهت مشخص شدن آخرین وضعیت دستگاه ها و لیست کردن مجدد آنها در زیر dev/) با این فرمان، دیگر نیازی نداریم mknod را به کار ببریم.

بررسی دایرکتوری proc/

این دایرکتوری، یک دایرکتوری خاصِ ساکن در حافظه و حاوی اطلاعات مختلف درباره کارهای درحال اجرا در سیستم لینوکس است .در زیر دایرکتوری proc مجموعه ای از دایرکتوری های عددی و تعدادی فایل مرتبط با تجهیزات وجود دارد.

$ cd /proc

$ ls -l | less

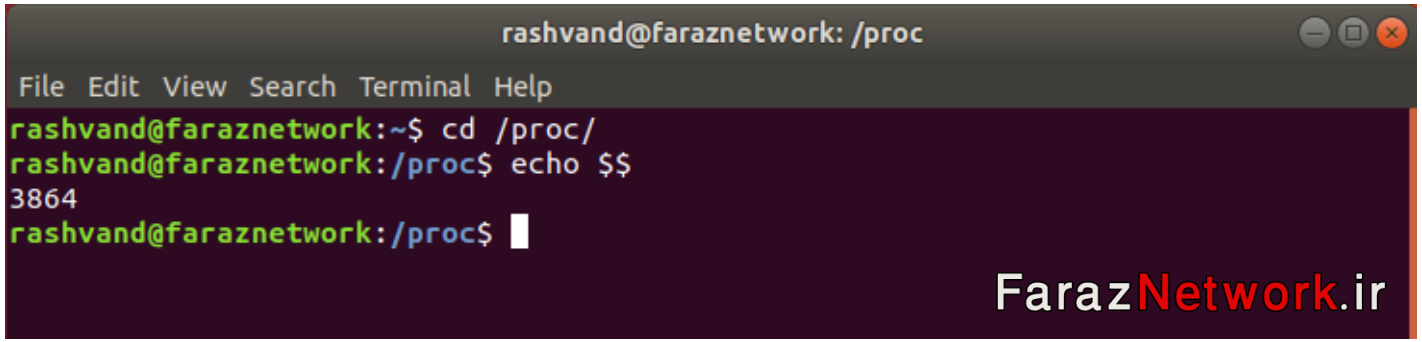


دایرکتوری proc فقط به صورت Logical است و موجودیت مستقلی نیست و آیینه ی تمام نما از اطلاعات Kernel است . با استفاده از داده های زیردایرکتوری /proc می توانید ببینید که کرنل چه فایل هایی را Open کرده و چند نفر login هستند. به محض اینکه سیستم را Down کنید دایرکتوری proc گم می شود و به محض اینکه Up کنید ساخته می شود. در زیر این دایرکتوری به ازای هر Process ی یک دایرکتوری داریم. وقتی فرمان زیر را در زیر دایرکتوری proc می زنیم به ما یک عدد می دهد، که شماره پروسس (شل) مرتبط به شما می باشد.

$ cd /proc

$ echo $$

3864

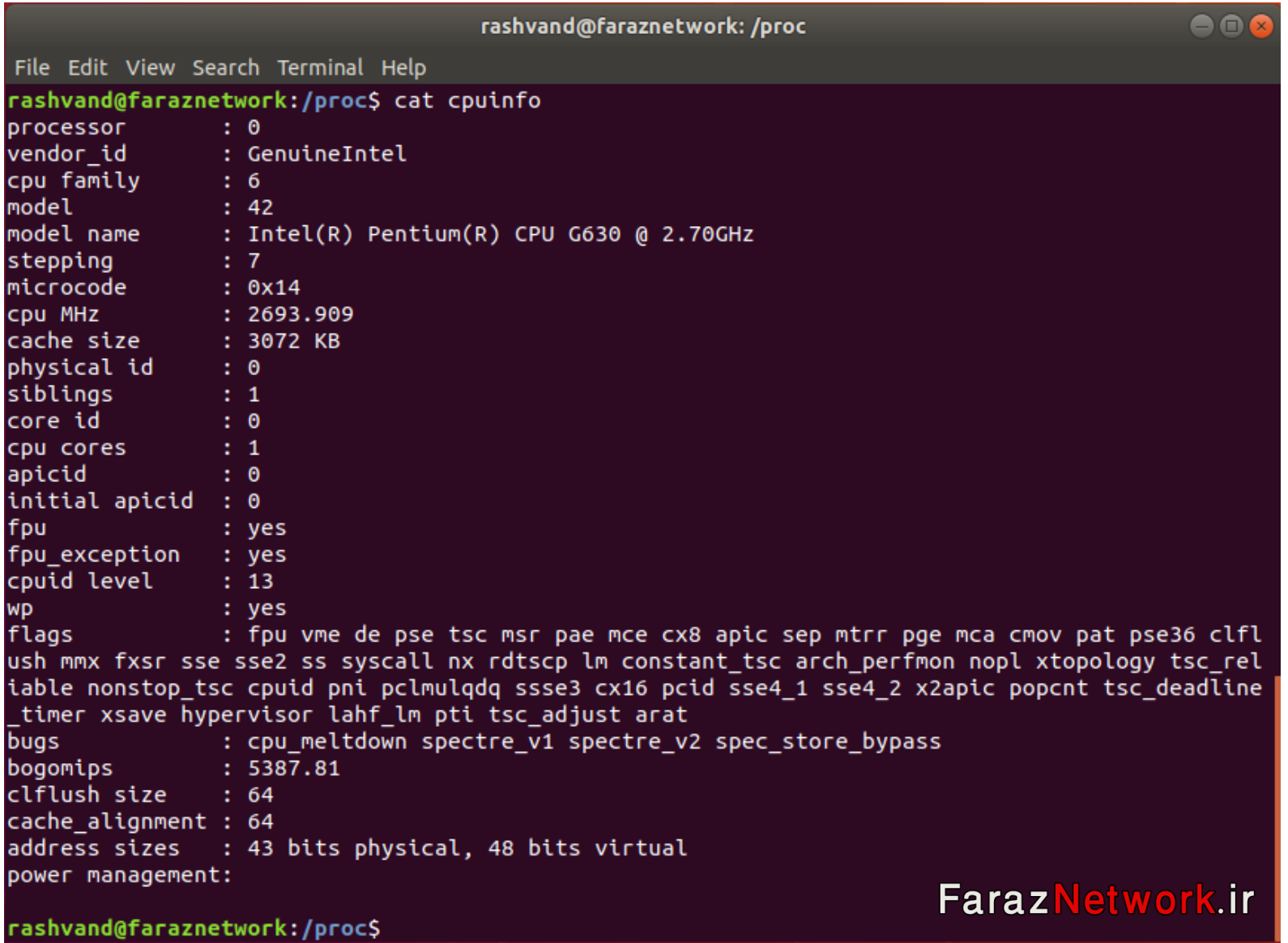


هر کاری که انجام دهید و هر فرمانی که بزنید Admin از طریق فایل های زیر دایرکتوری proc می تواند ببیند که کجا هستید. تعداد دایرکتوری هایی که زیر دایرکتوری proc وجود دارند ثابت نیست و پیوسته در حال تغییر است .

نکته : فرمان های who و w برای اینکه لیست کاربران Login کرده را تهیه کنند، از داده های زیر دایرکتوری proc استفاده می کنند.

البته در زیر این دایرکتوری تعدادی فایل مرتبط با تجهیزات نیز داریم. فرمان زیر را اجرا کنید :

$ cat cpuinfo



cpuinfo اطلاعات CPU را می دهد. ممکن است کسی از شما بپرسد که CPU این کامپیوتر از چه مجموعه دستوراتی (Instruction Set) پشتیبانی می کند؛ برای پاسخ به این سؤال کافی است به فایل /proc/cpuinfo مراجعه نمایید.

نکته: برای اجرای برخی از برنامه های کاربردی به خصوص در سطح سیستمی نیاز است که مجموعه دستورات خاصی توسط CPU پشتیبانی شود، که می توانید از خروجی فرمانcat /proc/cpuinfo در این مورد استفاده کنید.

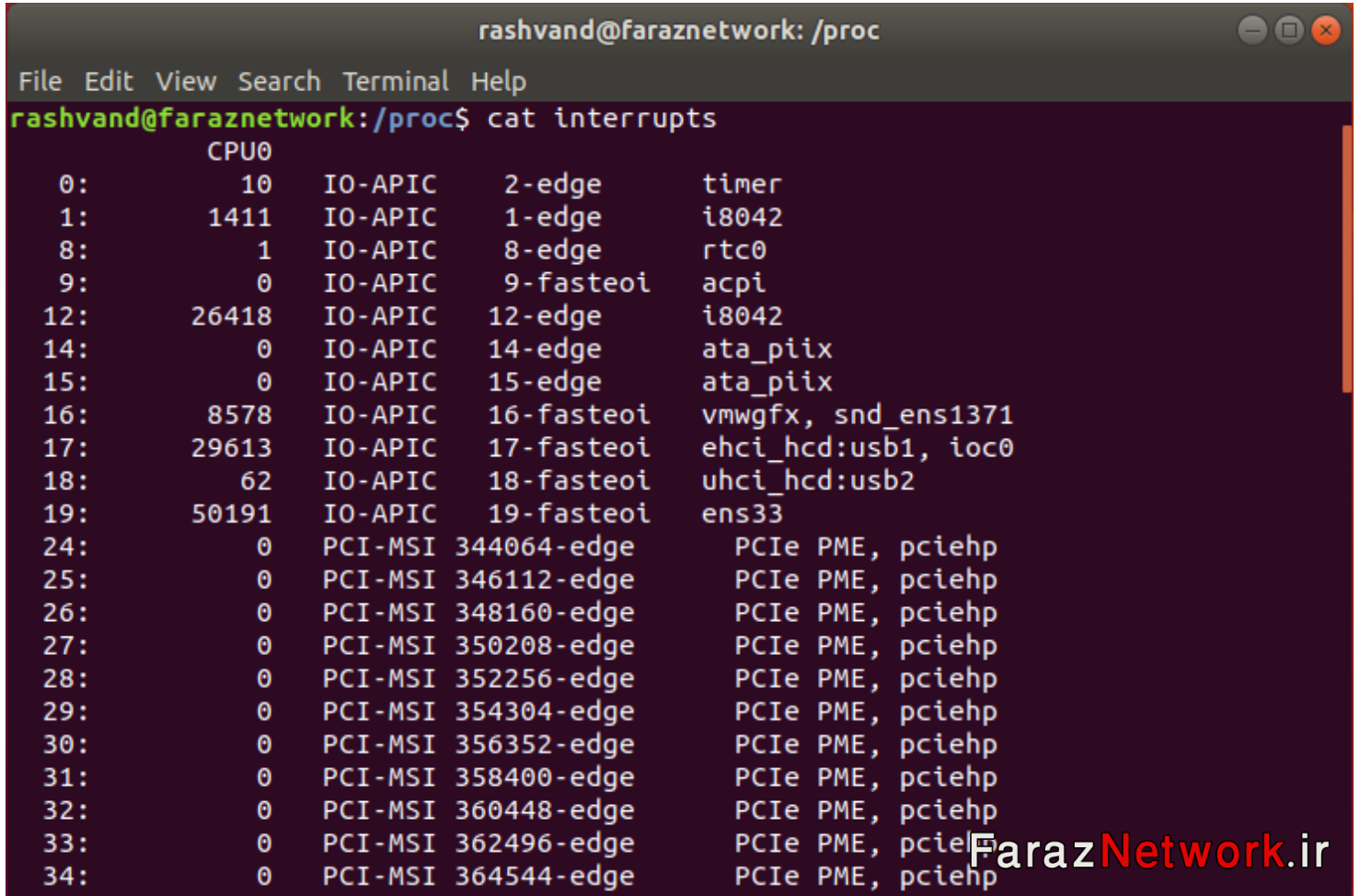
نکته : جهت دستیابی به اطلاعات دقیق تر در مورد CPU برنامه x86info را نصب کنید (apt-get install x86info یا yum install x86info) و سپس فرمان x86info -c را اجرا نمایید. خروجی این دستور اطلاعات بسیار دقیقی در مورد معماری پردازنده سیستم و معماری سیستمِ Cache آن را در اختیار شما می گذارد؛ این فرمان علاوه بر کاربرد در توسعه ی نرم افزارهای سیستمی، جهت خرید سیستم های حرفه ای و بررسی صحت اطلاعات موجود در کاتالوگ ها می تواند استفاده شود.

نکته : Bogomips (پردازش جعلی) : در پردازنده و زبان اسمبلی دستوری به نام nop(no operation) وجود دارد که فقط یک Cycle از CPU می برد، بدون اینکه کاری بکند و فقط زمانی را از پردازنده صرف خود می کند. مثلاً Device Driver دیسک می داند که اگر یک فرمان به دیسک بدهد باید نیم ثانیه بعد جواب بیاید به اندازه نیم ثانیه nop را انجام می دهد تا جواب بیاید. و اگر جواب نیامد Time out می دهد.

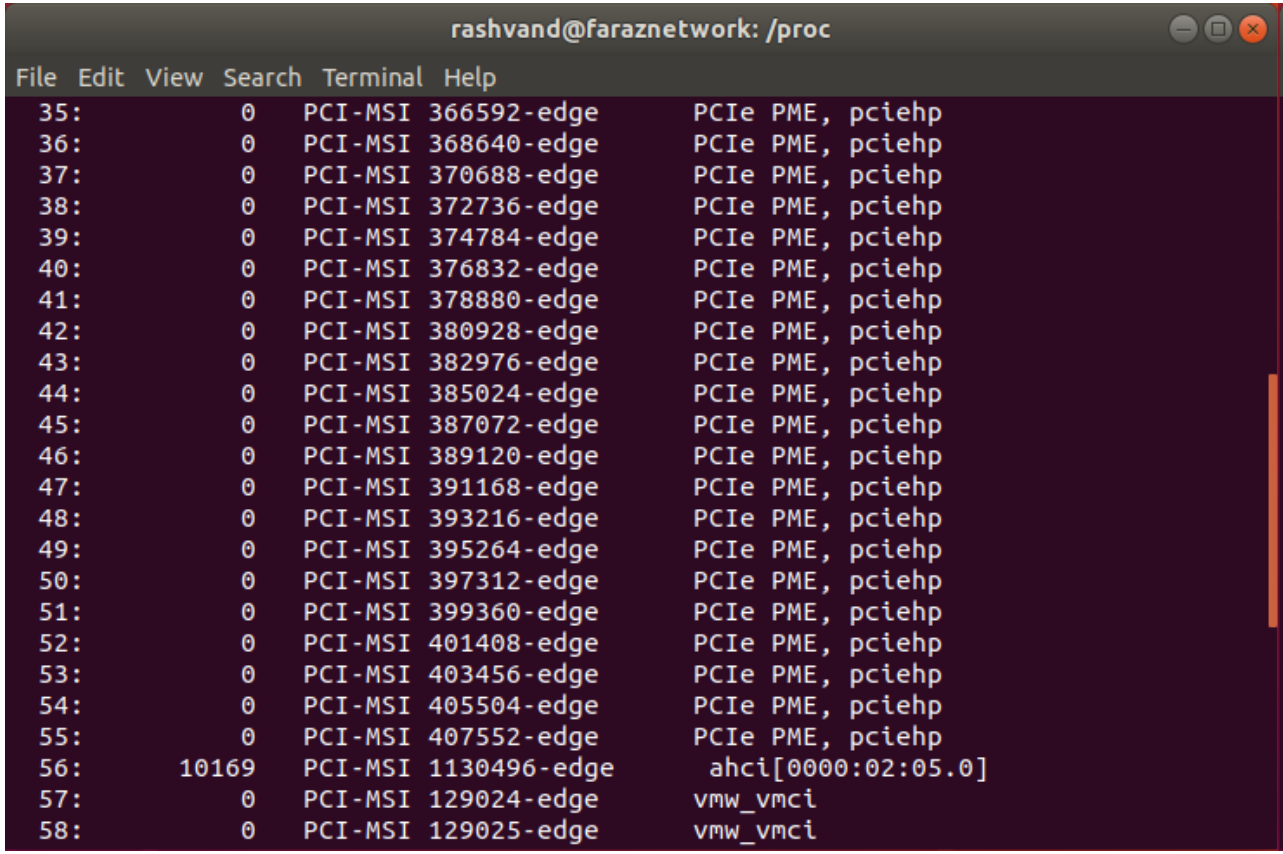
فایل interrupts

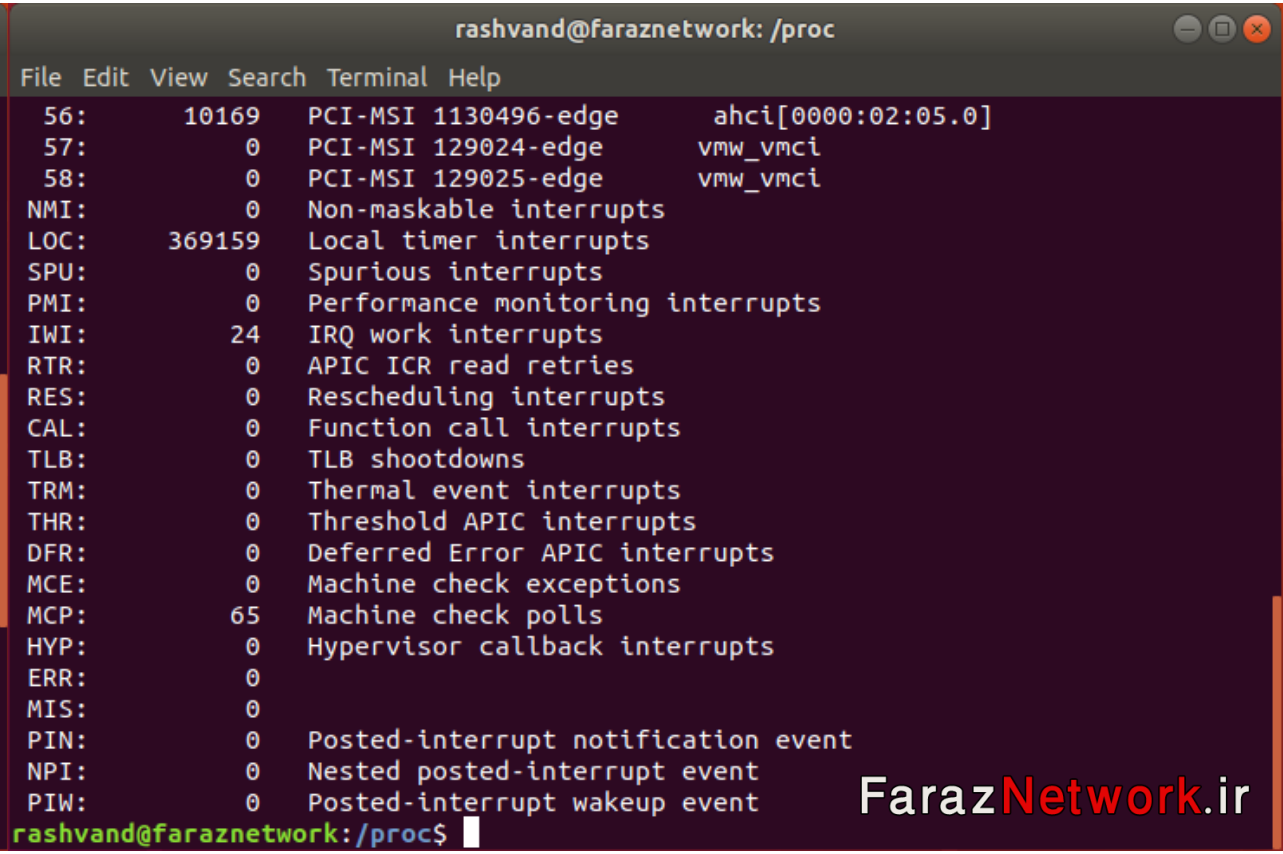
فایل بعدی که در زیر proc/ بررسی می کنیم interrupts است . در این فایل شماره هر interrupt و تعداد تکرار آن را می بینیم.

$ cat interrupts



هر مادربورد تشکیل یافته از تعدادی قطعه است که به یک بورد اصلی (یا بهتر است بگوییم یک بورد اصلی و چند بورد دیگر) وصل است . وقتی هر کدام از قطعه ها بخواهند با بورد مادر صحبت کنند Interrupt می دهند. همچنین دستگاه هایی هم که می خواهند با یکدیگر صحبت کنند Interrupt می دهد.

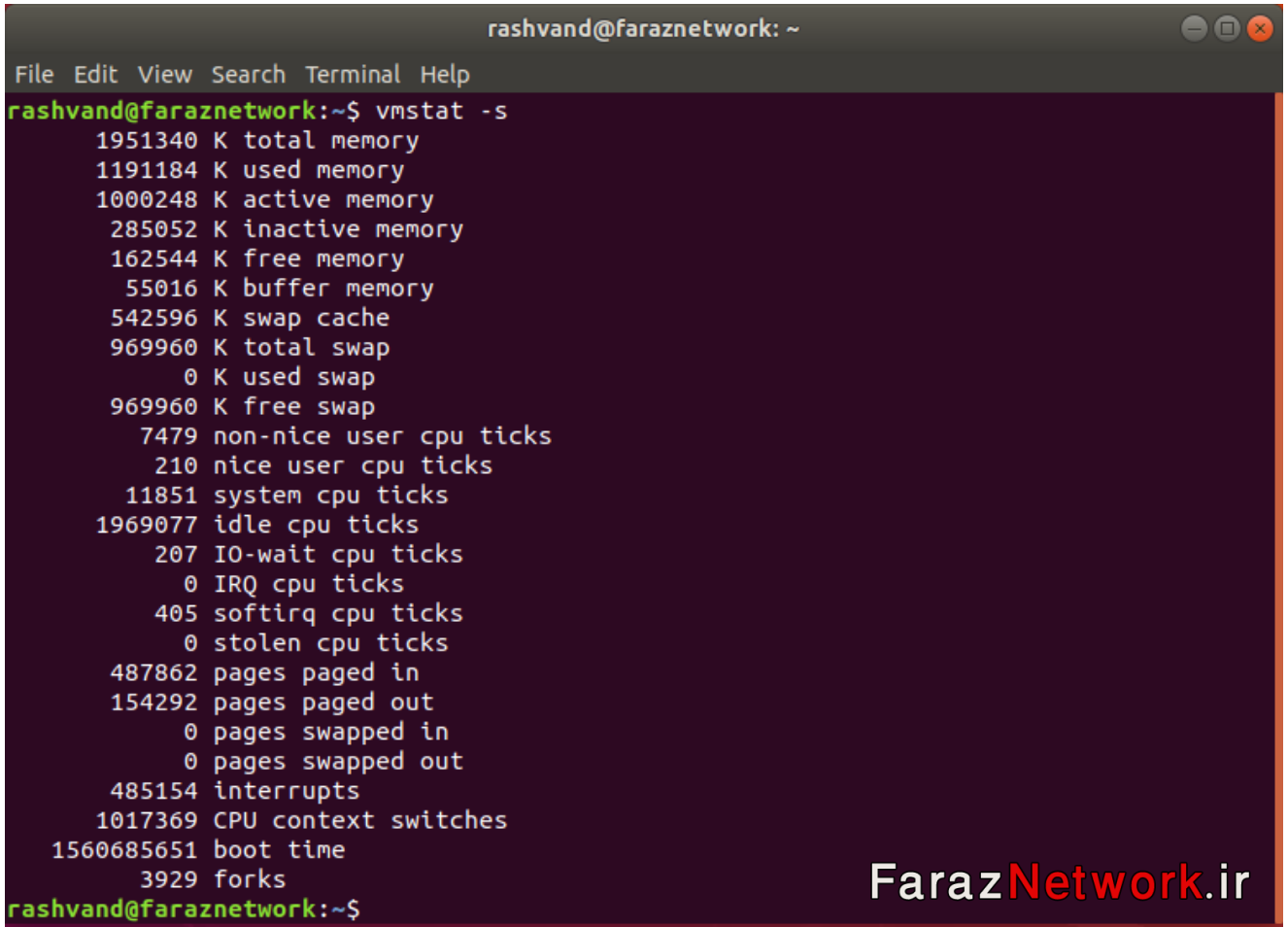




Timer Interrupt

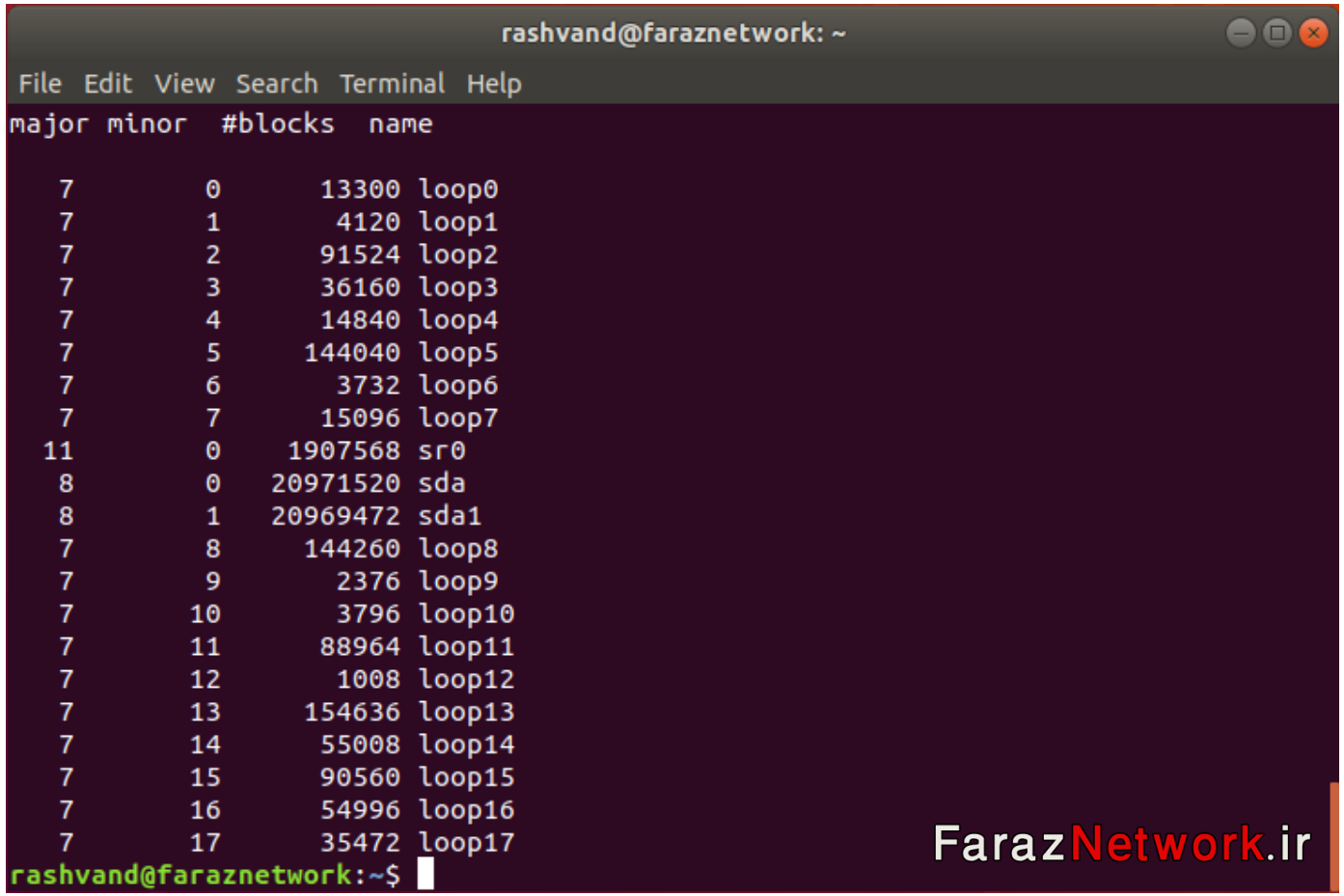
اگر فرمان vmstat -s را بزنیم در انتهای خروجی آن به ما می گوید که Context Switching چند بار انجام شده است . Context Switching به معنای واگذاری کنترل از یک Process به Process دیگر است.

$ vmstat -s



وقتی در دایرکتوری proc هستیم با فرمان زیر می توانیم مشاهده کنیم که هارد دیسک ما چند قسمت شده است :

$ cat partitions



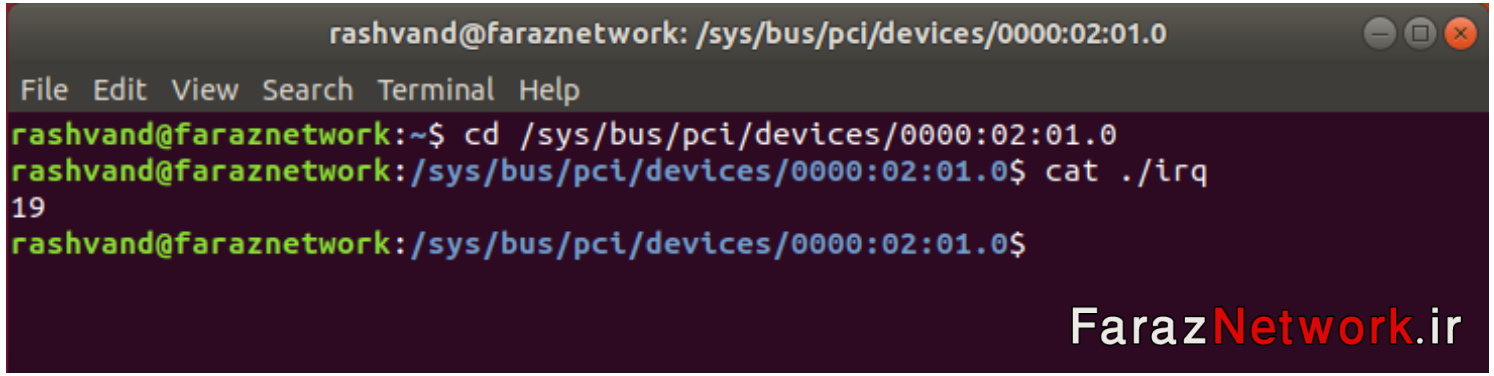
بررسی دایرکتوری /sys

فایل سیستمی مجازی در زیر دایرکتوری /sys قرار دارد که با نام sysfs شناخته می شود. این فایل سیستم محلی است که ابزارهای در سطح کاربر (و نه هسته) اطلاعاتشان را در مورد دستگاه های سیستم به دست می آورند. همچنین به واسطه ی این پوسته ی ارائه شده می توان تنظیمات مورد نظر را بر دستگاه های متصل به سیستم اعمال کرد (که البته معمولاً نیازمند دانش عمیق و تخصصی درمورد آن قطعه ی سخت افزاری است). مثلاً بر روی سیستم ما، همانطور که قبلاً دیدیم کارت شبکه در آدرس PCI با مقدار 02:01.0 قرار داشت . پس دستور زیر را اجرا می کنیم، که به دایرکتوری که حاوی فایل های مربوط به آن است وارد شویم:

```
$ cd /sys/bus/pci/devices/0000:02:01.0
```

حال فرض کنید که می خواهید بدانیم که شماره irq آن چند است . بعد از دستور بالا، دستور زیر را وارد می کنیم :

```
$ cat ./irq
```

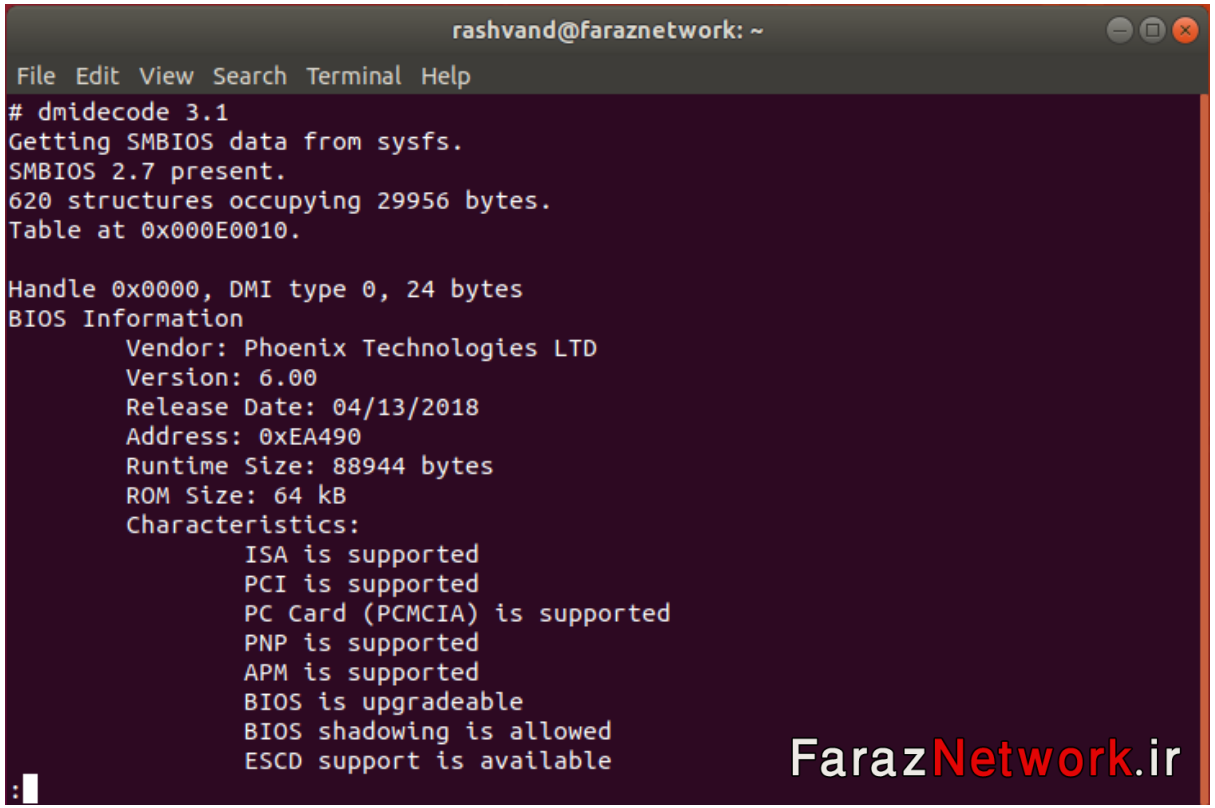


که در سیستم ما شماره ی ۱۹را بر می گرداند.

دستوراتی برای بررسی سخت افزار

زیر proc تعداد زیادی دایرکتوری عددی و فایل سخت افزاری داشتیم. تعدادی فرمان داریم که اطلاعات را از proc می خوانند. یکی از آنها فرمان dmidecode است که سخت افزار را زیر و رو میکند و اطلاعات را از زیر proc در می آورد.

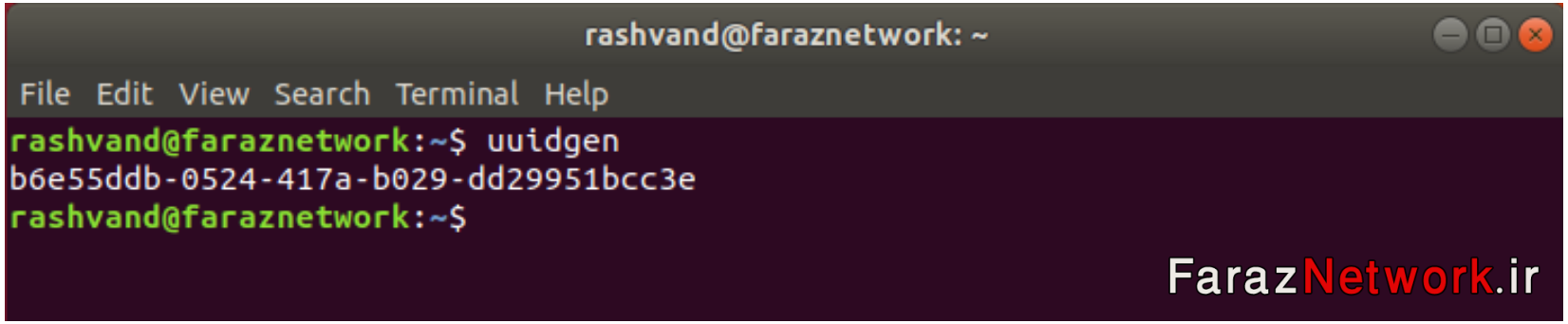
```
$ dmidecode |less
```



در خروجی دستور بالا اطلاعات مختلفی در مورد سخت افزار را مشاهده می کنید.

یک نکته ی مهم در مورد UUID یا "شناساگر یکتای جهانی"این است که وقتی Hard را فرمت می کنید UUID به شما می دهد. به عنوان مثال UUID کامپیوتر کلاس عدد ۳۲رقمی هگزا ۸.۴.۴.۴.۱۲ است .برای اینکه این عدد یکتا باشد و تکراری نبا شد، اوّل Mac Address کارت شبکه کامپیوتر را می گیرد و بعد از روی آن UUID را می سازد. اگر کامپیوتر شما کارت شبکه نداشته باشد cupid را می گیرد. در صورتی که خودتان خواستید UUID بسازید می توانید از فرمان زیر استفاده کنید:

$ uuidgen



پس یادتان باشد اگر خواستید Laptop بخرید چند فرمان را به یاد داشته باشید و CD Live لینوکس را ببرید و اطلاعات سخت افزاری آن را بررسی کنید.

علاوه بر فرمان dmidecode فرمان lshw نیز اطلاعات مهمی در مورد سخت افزار سیستم می دهد. برای مثال مشخصات کامل مادر بورد و باطری و موارد دیگر را در خروجی این دستورات می بینید. هرچند خروجی دستور dmidecode بسیار مفصل است ، در برخی مواقع، هرچند نادر، اطلاعات نادرستی را به نمایش می گذارد.

Coldplug در مقابل Hotplog

هر وقت با یک قطعه ی سخت افزار سر و کار داشتید باید یک نکته را در نظر بگیرید؛ اینکه این قطعه Coldplug است یا Hotplug .تفاوت این دو نوع سخت افزار در این است که قطعات Hotplug را چه در موقع خاموش و چه در موقع روشن بودن سیستم می توانید به سیستم متصل کنید یا از آن جدا کنید، ولی قطعات Coldplug را حتماً باید در موقع خاموش بودن سیستم وصل یا جدا کنید.

تذکر مهم : متصل یا جدا کردن قطعات سخت افزاری Coldplug در هنگامی که کامپیوتر روشن است ممکن است به آن قطعه یا کل کامپیوتر آسیب بزند. بنابراین هرگز چنین کاری نکنید و قبل از متصل یا جدا کردن قطعه ی سخت افزاری خاصی از سیستم از Coldplug یا Hotplug بودن آن اطمینان حاصل نمایید.

به طورمعمول قطعاتی مانند پردازنده، حافظه ی RAM، کارت های PCI (مثل کارت شبکه)، و هارد دیسک ها Coldplug هستند. هرچند نوع Hotplug آنها نیز برای کاربردهای خاص طراحی شده است . نوع Hotplug این قطعات عموماً برای سرورهایی که امکان خاموش کردن آنها وجود ندارد به کار برده می شوند. مثلاً در کامپیوترهایی که زیرساخت های تلفن همراه را پشتیبانی می کنند از RAM های Hotplug استفاده می شود. به این ترتیب در صورتی که یکی از RAMها خراب شود، فوراً می توان آن را تعویض کرد و در سیستم کار تلفن ها مشکلی پیش نمی آید.

قطعات سخت افزاری جدید خارجی (که نیازی نیست داخل Case باشند و از خارج کامپیوتر وصل می شوند)، مثل USB،Ethernetو -IEEE 1394 یا همان Firewire به صورت Hotplog هستند و در هنگام روشن بودن سیستم امکان اتصال و جدا کردن آنها وجود دارد.

نکته: به برنامه های معمولی که در حال اجرا هستند، خواه توسط یک کاربر معمولی یا Root، برنامه ی User-Space گفته می شود. در مقابل این برنامه ها، برنامه های Kernel-Space قرار دارند که بخشی از هسته ی سیستم عامل لینوکس هستند. برنامه های سمت هسته به طور مستقیم می توانند با سخت افزار در ارتباط باشند، اما برنامه های User-Space با استفاده از امکاناتی که Kernel فراهم می آورد، این کار را انجام می دهند.

HAL Daemon

HAL یا همان Hardware Abstraction Layer Daemon، یا HALD یک برنامه ی User-Space است که همواره در حال اجراست و به برنامه های User-Space اطلاعاتی در مورد قطعات سخت افزاری موجود می دهد.

D- Bus

BusDesktop همانند HALD یک Daemon است و همواره در حال اجراست . D-Bus این امکان را فراهم می آورد تا پروسس ها با یکدیگر در ارتباط باشند، همچنین در مورد Event ها (مثل متصل شدن یک قطعه ی سخت افزاری جدید از USB) هم از طریق سایر Process ها و هم از طریق سخت افزار کسب اطلاع کنند.

udev

از گذشته تا کنون لینوکس گره های مربوط به سخت افزار را در مسیر /dev ایجاد می کند. وجود قطعات سخت افزاری Hotplug باعث به وجود آمدن udev شد: یک فایل سیستم مجازی که در مسیر /dev ماونت (Mount) می شود و فایل های پویای دستگاه های سخت افزاری را ایجاد می کند. تنظیمات udev از مسیر /etc/udev انجام می شود. توجه داشته باشید که تنظیمات پیشفرض برای قطعات سخت افزاری رایج، کافی است .

اما قطعات سخت افزاری قدیمی تر مثل آنهایی که از پورت RS-232 یا Parallel وصل می شوند (مثلاً ماوس یا کیبورد های قدیمی یا چاپگرهایی که از پورت موازی وصل می شوند)، Coldplug هستند و لازم است که برای اتصال یا جدا کردن آنها کامپیوتر را خاموش کنید. یادتان باشد که اگر در موقع روشن بودن سیستم این کار را بکنید ممکن است به کامپیوتر یا آن دستگاه آسیب وارد شود.

تذکر: ممکن است شما در گذشته مانیتور را در حالتی که کامپیوتر روشن است جدا یا متصل کرده باشید و مشکلی پیش نیامده باشد، اما توصیه می شود که مانیتور را وقتی که کامپیوتر روشن است متصل یا جدا نکنید، زیرا امکان آسیب دیدن کامپیوتر یا مانیتور وجود دارد. توجه داشته باشید که استاندارد VGA چیزی در مورد hotpluggable بودن آن نمی گوید. البته در قطعات سخت افزاری جدید از دیودهای محافظ استفاده می شود، که در این صورت در صورت وجود آسیب، این دیودها خراب می شوند و قطعات داخلی سالم می مانند.

ادامه دارد ...