

انتقال کامل فایل سیستم از یک ماشین به ماشین دیگر

اگر می‌خواهید تمام سیستم فایل را از یک سیستم به سیستم دیگر منتقل کنید، مثلاً وقتی که کامپیوتر جدیدی تهیه کرده‌اید، این کارها را انجام دهید:

۱. هر دو کامپیوتر را با دیسک زنده لینوکس (مانند ناپیکس) بوت کنید و مطمئن شوید که از طریق شبکه همدیگر را می‌بینند.

۲. در ماشین مبدأ، پارتیشن‌های سیستم فایل‌ها را باید کپی شود، سوار (Mount) کنید و با استفاده از دستور `netcat` و `tar` عملیات انتقال را شروع کنید:

```
cd /mnt/sda1
tar -czpsf - . | pv -b | nc -l 3333
```

۳. در ماشین مقصد، پارتیشن‌های سیستم فایل را دریافت کند، سوار کنید و این کار را انجام دهید:

```
cd /mnt/sda1
nc 192.168.1.1
3333 | pv -b | tar -xpsf -
دستور (netcat) nc برای ارتباطات TCP به‌کار می‌رود. دستور pv برای نمایش روند پیشرفت انتقال. tar هم برای آرشیو کردن فایل‌ها در ماشین مبدأ و غیرفشرده کردن آن‌ها در ماشین مقصد انجام می‌شود.
```

سه تا پنج سال دیگر برای موفقیت اوبونتو

کانونیکال (Canonical)، پشتیبان تجاری سیستم عامل لینوکسی اوبونتو، هنوز به سود نرسیده است، اما بنیانگذار این شرکت، مارک شاتلورث در کنفرانسی تلفنی اعلام کرد که آماده است سرمایه شرکت را برای ۳ تا ۵ سال دیگر حفظ کند.

این فناوری پرست میلیون‌ها نفر است: «ما هنوز به سرمایه نیاز داریم و من از تک‌تک پنی‌هایی که دارم در این سرمایه‌گذاری‌ها می‌پردازم، نگرانم، اما به‌گمانم کار خوبی است. کانونیکال در کسب درآمد مثبت نبوده است، اما پیشنهاد ما برای آنهایی که می‌خواهند پنی‌های خود را ذره ذره در فضای لینوکس خرج کنند، پیشنهاد خیلی جذابی است.»

شاتلورث معتقد است که تنها راه درآمد در حیطه لینوکس، در آمد از راه ارائه خدمات است و نه فروش لینوکس، نبود سود در شرکت کانونیکال هم هنوز اعتقاد او به مدل تجاری‌اش را خدشه‌دار نکرده است، او معتقد است داریم به زمانی وارد می‌شویم که تحول واقعی سیستم‌های رومیزی است و دیگر از راه فروش سیستم عامل، نه تنها نمی‌توان پول زیادی به‌دست آورد، بلکه رفته‌رفته هیچ سودی عاید شرکت نخواهد شد.

Parallel SSH

```
$ mkdir example-tree
$ date > example-tree/df1.txt
$ date > example-tree/df2.txt
$ mkdir example-tree/subdir1
$ date > example-tree/subdir1/df3.txt
$ pscp -h hosts-file -l ben
```

```
example-tree/df1.txt /tmp/df1.txt
[1] 21:28:36 [SUCCESS] p1 22
[2] 21:28:36 [SUCCESS] p2 22
$ ssh p1 "cat /tmp/df1.txt"
Thu Oct 16 21:27:25 EST 2008
$ pscp -h hosts-file -l ben
example-tree /tmp/example-tree
...
```

```
python: Python/ceval.c:2918:
set_exc_info: Assertion
`frame != ((void *)0)' failed.
Aborted
$ pscp -h hosts-file -l ben --recursive
example-tree /tmp/example-tree
[1] 21:29:57 [SUCCESS] p1 22
[2] 21:29:57 [SUCCESS] p2 22
$ ssh p1 "ls -l /tmp/
example-tree"
```

```
total 24
ben 29 2008-09-25 16:01 df1.txt
-rw-r--r-- 1 ben
ben 29 2008-09-25 16:01 df2.txt
-rw-r--r-- 1 ben
drwxr-xr-x 2 ben ben 4096
2008-09-25
16:01 subdir1
```

نکته: `ben` نام کاربری فرضی است.

سیستمدستور `prsync` تنها چند تا از انتخاب‌های `rsync` را با خود آورده است. به‌همین خاطر، شما نمی‌توانید از `verbose` (ارائه گزارش عملکرد) در سیستم استفاده کنید. دستوری که در زیر می‌آید دایرکتوری `example-tree` را روی سرور همخوان می‌کند.

```
$ prsync -h hosts-file -l ben -a
--recursive example-tree /tmp
```

پوسته امن موازی یک دستور در چند سیستم

```
[1] 21:13:00 [SUCCESS] p2 22
p2:
Thu Sep 25 15:44:36 EST 2008
p1:
[2] 21:13:00 [SUCCESS] p1 22
p1:
```

معمولاً خروجی استاندارد میزبان‌های راه دور به شما نشان داده نمی‌شوند. انتخاب `-P` در آخرین تعامل، خروجی را در هر دو میزبان راه دور نشان می‌دهد. اگر شما از دستورات پیچیده‌تری استفاده می‌کنید، احتمالاً بخواهید از `-i` استفاده کنید تا خروجی‌های میزبان‌ها را گروه‌بندی شده ببینید. همچنین می‌توانید از `outdir` استفاده کنید تا مسیر پوشه‌ای را که می‌خواهید برای ذخیره خروجی هر میزبان راه دور استفاده شود، اعلام کنید. خروجی هر میزبان در فایل جداگانه‌ای همان‌با ماشین میزبان آن ذخیره می‌شود. می‌توانید از `--timeout` استفاده کنید تا نشان دهید که یک دستور چقدر می‌تواند طول بکشد. به طور پیش فرض این عدد ۶۰ ثانیه است. این بدین معنی است که اگر دستور شما طی ۶۰ ثانیه به پایان نرسید، `pssh` آن را خطا تلقی کرده و پیغام آن را بر می‌گرداند، که در زیر می‌بینید. می‌توانید زمان اجرای هر دستور را افزایش دهید تا از این خطا جلوگیری کنید.

```
# pssh -h hosts
-file -l ben -i "sleep 65; date"
[1] 21:19:26
[FAILURE] p1 22 Timeout
[2] 21:19:26 [FAILURE] p2
22 (4, 'Interrupted system call')
دستور pscp همان -h و -l و --timeout را می‌گیرد و شامل --recursive هم می‌شود تا از کپی عمیق از میزبان محلی نیز پشتیبانی کند. در ادامه این دستور، شما مسیرهای محلی و راه دوری که می‌خواهید فایل‌تان را کپی کنید، انتخاب می‌کنید، اولین دستور pscp در مثال زیر، یک فایل را روی دو ماشین راه دور به‌طور موازی ارسال می‌کند. دستور ssh‌ای که به دنبال آن می‌آید مشخص می‌کند که آن فایل در p1 وجود دارد یا خیر. با دانستن این‌که می‌خواهیم یک دایرکتوری را کپی کنیم، می‌توانیم با افزودن --recursive از بروز خطا در سیستم جلوگیری کنیم.
```

محمد رضا قربانی

بسیاری از ما از `SSH` برای ورود از راه دور به ماشین‌ها استفاده می‌کنیم. فایل کپی می‌کنیم و کارهای عمومی مدیریت سیستم را انجام می‌دهیم. اگر می‌خواهید کارایی‌تان را با `SSH` بالا ببرید، می‌توانید از ابزاری استفاده کنید که به شما اجازه می‌دهد کارایی‌تان را افزایش دهید و یک دستور را به‌طور همزمان در چند ماشین اجرا کنید. `ssh`، `Parallel SSH`، `Cluster SSH` و `ClusterIt` به شما اجازه می‌دهند دستورات را در یک پنجره پایانه وارد و به چند ماشین مختلف ارسال کنید.

چرا به چنین ابزاری نیاز دارید، در حالی که با استفاده از `openSSH` می‌توانید فایل‌های دستورات خود ایجاد و در آن از دستور حلقه استفاده و در چند سیستم مختلف اجرا کنید؟ خب، اگر این دستورات بخواهند همزمان در چند ماشین اجرا شوند، یا اگر بخواهیم در بلندمدت، وظایفی را اجرا کنیم که زمانبر هستند، یا یک فایل را در چند سیستم ویرایش کنیم، در این صورت، دستورات موازی بسیار سریع‌تر و کارآمدتر خواهند بود.

Parallel ssh (pssh)

پروژه `pssh` شامل نسخه‌های موازی‌ای از پوسته‌های `(pscp)`، `(scp)`، `(prsync)` و `rsync` و `(pnuke)` است.

`Pssh` برای `openSUSE` آماده شده و در کتابخانه یونیورس اوبونتو موجود است. تمام دستورات `ssh` موازی شکل و ساختار `hosts-file` `-h` command را دارند، که `hosts-file` لیستی از میزبان‌هاست که می‌خواهید دستور در آن اجرا شود. مثلاً اولین دستور زیر تاریخ را در `p1` و `p2` اجرا می‌کند. آرگومان `-l` نام کاربری را مشخص می‌کند که بایستی برای ورود به ماشین راه دور استفاده کرد.

```
# cat hosts-file
p1
p2
# pssh -h hosts-file -l ben date
[1] 21:12:55 [SUCCESS] p2 22
[2] 21:12:55 [SUCCESS] p1 22
# pssh -h hosts-file -l ben -P date
Thu Oct 16 21:14:02 EST 2008
p2:
```

اینتل در استفاده از سیستم عامل لینوکسی به تایوان پیوست

اینتل و دولت تایوان قرار است توسعه‌ای ایجاد کنند که در آن سیستم عامل مبتنی بر لینوکس مابلین (Moblin) را برای دستگاه‌هایی چون نت‌بوک‌ها و اینترنت موبایل (MID) که از میکروپروسسورهای سری اتم اینتل استفاده می‌کنند، توسعه دهند.

اینتل و دولت تایوان قرار است توسعه‌ای ایجاد کنند که در آن سیستم عامل مبتنی بر لینوکس مابلین (Moblin) را برای دستگاه‌هایی چون نت‌بوک‌ها و اینترنت موبایل (MID) که از میکروپروسسورهای سری اتم اینتل استفاده می‌کنند، توسعه دهند.