

## Configurando um sistema de impressão

# LPRng - A nova geração

O servidor de impressão mais popular no mundo \*NIX é o lpd, utilizado desde tempos longínquos. Seu sucessor, o LPRng, segue a mesma filosofia (e forma de configuração), mostrando-se um sistema confiável, leve e rápido para gerenciar e utilizar impressoras. **POR PITER PUNK**



Yuri Maslov - www.sxc.hu

A filosofia UNIX dita o uso de vários pequenos programas, cada qual cuidando muito bem de uma função específica, trabalhando em conjunto para realizar grandes feitos. Um bom exemplo disso é o sistema de impressão no Linux.

No nível mais baixo, temos o kernel, que trabalha diretamente na comunicação com a impressora. É nesse nível que vamos encontrar o suporte à porta paralela, serial, USB e à rede. Obviamente, se você pretende utilizar uma impressora que se conecta à porta paralela, irá precisar do suporte a esse tipo de dispositivo no kernel. A maioria das distribuições Linux já fornece este recurso, se não compilado no kernel, como um módulo carregado automaticamente sempre que necessário.

Logo acima do kernel, encontramos o filtro de impressão. Este programa irá traduzir o documento a ser impresso para a linguagem específica da impressora. Sim, as impressoras possuem uma linguagem própria e os documentos devem ser traduzidos para esse idioma que, via de regra, muda de fabricante para fabricante e, dentro de um mesmo fabricante, de modelo para modelo. O filtro cuida disso, convertendo os documentos recebidos para algo que a impressora entenda.

Quem recebe o documento e o envia para o filtro de impressão é o daemon de impressão. Pode ser o BSD LPD, o CUPS ou o LPRng, entre outros. A função destes daemons é receber o trabalho enviado pelo usuário e enviá-lo para a impressora correta através do filtro cor-

reto. Se for uma impressão local, ele deve receber o documento, passá-lo pelo filtro e enviar o resultado para o dispositivo de impressão. Caso seja uma impressora remota, ele recebe o documento e conversa com o daemon da máquina remota, enviando-o para ela. Algumas vezes, o filtro é executado na máquina de origem do trabalho e, em outras, na máquina de destino.

O último elo dessa cadeia são os comandos do usuário, que enviam os trabalhos para o daemon e recolhem dele informações sobre o andamento da fila de impressão e o que está acontecendo com cada trabalho.

Neste artigo iremos configurar uma impressora utilizando o LPRng. Veremos quais os arquivos envolvidos, os parâmetros mais importantes e criare-

mos um pequeno filtro de impressão. Existem ferramentas que facilitam e automatizam boa parte desse processo, mas nenhuma delas oferece todos os recursos e tampouco a flexibilidade que o conhecimento do sistema como um todo pode trazer.

## No princípio...

Era o vazio, e os usuários mandavam seus dados diretamente para os dispositivos. O *cat* reinava soberano. Para enviar um documento para a impressora bastava ordenar:

```
# cat documento > /dev/lp
```

E o documento era impresso. As impressoras daquela época lidavam apenas com texto, mas mesmo assim podiam estar em outra máquina, e por isso existiam diferentes formas de imprimir. Para resolver esses problemas surgiu o *lpd*, que lê no arquivo */etc/printcap* onde está a impressora e qual filtro de impressão deve ser usado.

## Nossa primeira impressora

É extremamente simples configurar uma impressora no BSD LPD ou no LPRng. Basta colocar os dados referentes à ela no arquivo */etc/printcap*. A seguir, um exemplo de como é a entrada de uma impressora neste arquivo:

```
lp|printer1:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:
```

A primeira coisa que temos é o nome da impressora (*lp*), seguido do apelido dado a ela (*printer1*), separado pelo caractere *pipe* (*|*). A partir daí, temos o restante da configuração, com cada um dos campos separado por “:”

- *lp=/dev/lp0*: aqui definimos qual o dispositivo da impressora. */dev/lp0* é a primeira porta paralela.
- *sd=/var/spool/lpd/lp*: este é o diretório no qual vai ficar o arquivo de impressão, enquanto o *daemon* espera a impressora estar pronta para imprimir. Imagine a cena: vários usuários mandam imprimir diferentes arquivos e a impressora só pode imprimir um de cada vez. Enquanto

um dos arquivos é impresso os outros ficam esperando na fila, que costumamos chamar de *spool*. Quando a impressora termina de imprimir o primeiro trabalho, ela passa para o próximo da fila.

- *mx#0*: esse misterioso comando define o tamanho máximo do arquivo a ser impresso. Se não quiser impor nenhum limite, use o 0 (zero).
- *sh*: essa linha, que não é mais necessária no LPRng, avisa ao *daemon* que não deve ser impressa uma página de “rosto” com o nome de quem imprimiu as páginas seguintes, nome do arquivo e outras informações. O comportamento padrão do BSD LPD é imprimir essa folha de rosto, por isso a necessidade dessa opção. No LPRng o padrão é não imprimir.

Pronto, já temos a nossa primeira entrada de impressão. Agora devemos avisar ao *daemon* que nossa nova impressora existe. Primeiro, usamos o comando *checkpc*:

```
# checkpc -f
```

Este comando irá criar todos os arquivos de que nossa nova impressora precisa. Lembre de executá-lo sempre que adicionar uma nova impressora. Agora vamos notificar o *lpd*:

```
# lpc reread
```

Com esse comando avisamos ao *daemon* para ler novamente seus arquivos de configuração. Agora a nossa impressora já está pronta para uso.

Infelizmente nem tudo é simples como deveria ser. Para que funcione com apenas as definições acima, uma impressora deve entender arquivos texto no formato UNIX e arquivos PostScript. Normalmente elas apenas compreendem arquivos no formato DOS e usam um formato exótico qualquer para imprimir gráficos (como o PCL, da HP). Podemos resolver esse pequeno problema adicionando um filtro à nossa definição de impressora, como veremos mais adiante ainda neste artigo.

## Imprimindo em rede...

Configurar uma impressora de rede é tão simples como uma impressora local; aliás, é até mais simples. De maneira

geral, para economizar trabalho, o filtro de impressão é instalado na máquina na qual está conectada a impressora, ou seja, não precisamos nos preocupar com ele. Uma entrada para uma impressora remota se parece com a seguinte:

```
lp1|remote-printer:\
:rp=lp:\
:rm=maquina-remota.lalala:\
:sd=/var/spool/lpd/lp1:\
:sh:
```

Bem semelhante à entrada anterior, não acham? A grande diferença são os campos *rp* e *rm*. O primeiro é o nome da impressora na máquina remota (no exemplo dado, *lp*) e o segundo é a máquina na qual a impressora está conectada (no exemplo, *maquina-remota.lalala*). Esta é a sintaxe tradicional do BSD LPD. No LPRng as coisas são ainda mais simples:

```
lp1|remote-printer:\
:lp=lp@maquina-remota.lalala:\
:sd=/var/spool/lpd/lp1:
```

O campo *:sh*: não aparece porque, como já foi dito, ele não é necessário no LPRng. Os campos *rm* e *rp* são mesclados em um único campo *lp*, onde informamos a impressora e a máquina à qual ela está conectada.

Com isso já sabemos configurar impressoras locais e impressoras remotas, bastam apenas algumas linhas no arquivo */etc/printcap*.

## Uma fila para várias impressoras

Uma situação comum em ambientes corporativos é termos várias impressoras cumprindo a mesma função. Com isso, podemos ter uma impressora “mestre” que repassa os trabalhos às escravas à medida que elas forem desocupadas. Há duas vantagens nesse método:

1. Você vai usar melhor suas impressoras, já que não vai ficar com uma delas entupida de trabalho e outra parada.
2. Não precisa configurar *N* impressoras em todas as máquinas da sua rede. Configure só a impressora mestre e cuide das outras direto no servidor.

Montar uma configuração deste tipo usando o LPRng é fácil. Veja um exemplo a seguir:

```
master:\
:sd=/var/spool/lpd/%P:\
:sv=slave1,slave2,slave3:
```

Duas novidades, o uso do parâmetro `%P` e o campo `sv`. O `%P` apenas diz ao LPRng que o nome do spool é o mesmo da impressora (no nosso caso, `/var/spool/lpd/master`). Note que a cada exemplo vamos deixando isso cada vez mais simples.

Já a linha `sv` descreve quais são as impressoras “escravas”; `sv` significa *slave*. Chamei as impressoras escravas de `slaveN`, onde `N` é um número de 1 a 3. A configuração das impressoras escravas é idêntica à de uma impressora normal, exceto por um campo:

```
slave1:\
:lp=/dev/lp0:\
:sd=/var/spool/lpd/lp1:\
:mx#0:\
:ss=master:
```

Note a linha `ss=master`, que indica o nome da impressora mestre à qual essa escrava serve. Você deve incluir esse campo na configuração das impressoras escravas. Em seguida, execute os comandos `checkpc` e `lpc reread` para fazer as suas novas impressoras funcionarem.

## Filtros...

Quando o assunto são os filtros, as coisas começam a ficar um pouco mais complicadas - e quase sempre precisamos lidar com eles. No caso de impressoras remotas, nem sempre isso é necessário. Impressoras de rede costumam compreender o formato PostScript e, no caso de impressoras conectadas a outra máquina \*NIX, o filtro geralmente já está instalado na ponta “de lá”.

Mas existem as impressoras locais e nem sempre os filtros estão realmente instalados na máquina remota. É necessário um filtro diferente para cada tipo de impressora, às vezes até mais que um. Adicionar um filtro a uma de nossas entradas no `printcap` é fácil:

```
lp|printer1:\
:lp=/dev/lp0:\
:if=/caminho/para/o/filtro:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:
```

Ao comparar com a entrada que temos na seção anterior, você verá que a única diferença é o campo `if`, onde está especificado o caminho completo para o filtro. O filtro é específico para um determinado modelo de impressora, no caso a Epson Stylus C42SX. O exemplo abaixo é voltado a ela:

```
#!/bin/sh
#
/usr/bin/gs -q -dSAFER -dBATCH >
-dNOPAUSE -dPARANOIDSAFER >
-sDEVICE=ijs >
-sIjsServer=/usr/bin/ijsgimpprint >
-dIjsUseOutputFD -sPAPERSIZE=a4 >
-sDeviceManufacturer=EPSON >
-sDeviceModel=escp2-c40sx >
-sOutputFile=- -
```

O texto acima, que começa com `/usr/bin/gs` e termina com `-`, é escrito todo em uma única linha, sem quebras. Vamos dissecar esse monstro para ver como ele funciona:

- `/usr/bin/gs` é o GhostScript, nosso “superfiltro”. Ele já vem com vários filtros embutidos, além de permitir a adição de filtros novos através de módulos especiais.
- `-q -dBATCH -dNOPAUSE`: esses três comandos fazem com que o GhostScript não gere mensagens (modo silencioso, `-q`), que não haja uma pausa após a tradução de cada página (`-dNOPAUSE`) e que o prompt de comando do GhostScript não seja mostrado no final do trabalho (`-dBATCH`). Esses passos são importantes para que nada atrapalhe a impressão do nosso documento.
- `-dSAFER -dPARANOIDSAFER`: essas opções de segurança são utilizadas para garantir a integridade dos seus arquivos e evitar que algum código PostScript faça algo errado no seu sistema. Sim, PostScript é uma linguagem de programação e pode ser usada para fazer as coisas mais estranhas, como servidores Web.
- `-sDEVICE=ijs` escolhe o tipo de dispositivo a ser utilizado. Quando utilizamos o tipo `ijs`, preparamos o GhostScript para carregar um filtro externo. No exemplo, estamos usando os filtros do `gimp-print`.
- `-sPAPERSIZE=a4` informa o tamanho de papel. Usamos o formato A4.

- `-sOutputFile=-` configura o arquivo de saída para a saída padrão. O LPRng irá cuidar de receber a saída padrão e direcioná-la para a impressora.
- `-` simboliza a entrada padrão, que é por onde o nosso filtro de impressão está recebendo o arquivo.
- `-sIjsServer=/usr/bin/ijsgimpprint`: essa opção indica ao GhostScript quem será o responsável pelo filtro. Além desta, temos também outras opções específicas do servidor IJS.
- `-dIjsUseOutputFD` diz ao servidor IJS para ser bonzinho e obedecer aos nossos redirecionamentos para a saída padrão (normalmente ele se recusa a fazer isso).
- `-sDeviceManufacturer=EPSON -sDeviceModel=escp2-c40sx` especifica o fabricante e o modelo da impressora. É aqui que especificamos o filtro propriamente dito. Para ver uma lista com todas as impressoras suportadas pelo `gimp-print`, consulte a página de manual do `gimp-print-models`.

Vamos chamar este filtro de `myfilter`, transformá-lo em um arquivo executável (`chmod +x myfilter`) e colocá-lo no diretório `/usr/local/bin`. Nossa entrada de impressora vai ficar assim:

```
lp|printer1:\
:lp=/dev/lp0:\
:if=/usr/local/bin/myfilter:\
:sd=/var/spool/lpd/lp:\
:mx#0:\
:sh:
```

Infelizmente o trabalho ainda não acabou. Da maneira como fizemos, o filtro só consegue trabalhar com arquivos no formato PostScript. Embora a maior parte dos programas gráficos do mundo \*NIX gere saída em PostScript, um filtro do mundo real tem que trabalhar com diversos formatos de arquivo, inclusive texto puro. Para isso, é necessário descobrir qual o tipo do arquivo que está sendo impresso.

Podemos fazer isso em nosso filtro, mas já existem diversos filtros prontos que incluem este e outros recursos. O melhor a fazer é utilizar algum programa auxiliar, como o `apsfilter`, que irá guiá-lo por uma série de menus em modo texto até que sua impressora esteja devidamente configurada. Se por acaso sua impressora não for suportada, ou se você

estiver planejando fazer algum uso mais “criativo” do filtro (como reduzir todas as páginas e imprimir duas por folha, etc...) já tem uma idéia do que fazer.

### Controlando tudo isso: lp\*

Depois de criar nosso filtro e configurar as filas de impressão, precisamos colocar tudo isso para funcionar. Afinal, não adianta nada termos impressoras configuradas e não imprimirmos nada. Os principais comandos usados para controlar o LPRng e enviar trabalhos para as impressoras são:

- *lpr* - Envia um trabalho (arquivo) para a impressora
- *lpq* e *lpstat* - Verifica o estado da fila de impressão
- *lprm* - Remove um trabalho da fila
- *lpc* - Controla o sistema de impressão

Vamos analisar cada um destes comandos, em detalhes, a seguir.

### lpr

O comando mais importante, e com certeza o mais usado, é o *lpr*. Com ele podemos mandar um trabalho para uma impressora, seja ela local ou remota. A

forma de uso é bem simples. Para imprimir um arquivo digite:

```
# lpr arquivo
```

ou podemos enviar o arquivo a ser impresso para a fila via pipes.

```
# cat arquivo | lpr
```

Nos dois casos, o “arquivo” será enviado para o daemon de impressão e, no momento oportuno, para a impressora padrão. No BSD LPD, qualquer impressora com o nome *lp* será a impressora padrão. Já no LPRng as coisas são diferentes: a impressora padrão será a primeira entre as listadas no arquivo */etc/printcap*. Por pura tradição, geralmente o nome dela também é *lp*.

Existem outras maneiras de determinar a impressora padrão; uma das mais simples é colocá-la na variável de ambiente `PRINTER`. Assim, se quisermos que a impressora padrão seja a *lp\_aux*, devemos digitar o comando:

```
# export PRINTER=lp_aux
```

Você pode colocar essa variável de ambiente no seu *.bashrc* ou *.bash\_profile*. Também é possível especificar a impressora diretamente na linha de comando:

```
# lpr -Plp_aux arquivo
```

Ou, indo mais além, especificar uma impressora remota diretamente:

```
# lpr -Plp@outra_maquina arquivo
```

Além de enviar um trabalho para a impressora, o *lpr* possui diversas outras opções; veja algumas delas:

- *-C letra* determina a prioridade da impressão. A menor é *A* e a maior é *Z*. A configuração padrão envia todos os trabalhos como tendo a prioridade mais baixa. O *C* significa “class”.
- *-B* junta vários trabalhos em um só. Assim, você pode mandar imprimir vários arquivos como sendo apenas um trabalho. Útil para ajudar no gerenciamento dos trabalhos.
- *-Kn* imprimem cópias do trabalho enviado. É um recurso muito útil, porém pouco conhecido.

- *-b*: este parâmetro habilita o modo “binário”, no qual o sistema de impressão tenta manter o arquivo “intocado”. Muito útil quando utilizado em conjunto com uma impressora *raw* (sem filtro). Dessa maneira, o sistema de impressão se comporta praticamente como um *pipe*, enviando o trabalho diretamente da máquina cliente para a impressora.

## lpq

Apesar dos nomes diferentes, *lpq* e *lpstat* são praticamente o mesmo comando, sendo o *lpstat* uma espécie de front-end para o *lpq*, com sintaxe e algumas opções diferentes. O objetivo desses comandos é verificar o estado de uma fila de impressão, se está ocupada, quem está imprimindo, se ela existe, entre outras informações. Vamos explicar como funciona o *lpq*, pois com ele é possível coletar todas as informações que desejamos. Caso precise de alguma informação adicional, consulte as páginas de manual dele e do *lpstat*. O comando básico é:

```
# lpq -Pall -s
```

Esse comando vai lhe mostrar todas as impressoras configuradas na máquina em que você está e quantos trabalhos estão na fila de cada uma delas. Uma saída típica seria:

```
# lpq -Pall -s
master@rachael 0 jobs (subservers lp)
lp@rachael 0 jobs (servng master)
mp3@rachael 0 jobs
lp1@rachael 0 jobs
```

O *-s* é de “short”; sem essa opção são apresentadas várias informações extras. O parâmetro *-P* funciona de maneira idêntica ao do comando *lpr*. A diferença aqui é o *all*, que é um apelido indicando todas as impressoras. Se quiser fazer a mesma pesquisa em uma máquina remota, basta digitar:

```
# lpq -Pall@maquinaremota -s
```

Com isso, mostramos as informações das filas de impressão configuradas na máquina remota.

Retirando o *-s*, temos várias outras informações sobre fila de impressão, sendo a principal a lista dos trabalhos sendo impressos. Uma saída típica inclui, entre outras informações, as mostradas no Quadro 1. Com isso podemos saber que:

1. Existe um trabalho sendo impresso (active), ele pertence ao usuário punk, é da classe A (a mais baixa), possui o número 187 e está sendo impresso desde as 20:35:53

2. O tal usuário punk também imprimiu um trabalho da mesma classe as 12:31:32. Diga-se de passagem, um arquivo bem maior, com quase 36k

Estas são informações bem interessantes; podemos obtê-las em várias impressoras ao mesmo tempo e, com isso, ter a qualquer momento um instante do que está acontecendo no sistema de impressão.

## lprm

Esse é um comando simples de usar, mas que geralmente tem pouca utilidade. Normalmente, quando percebemos que estamos imprimindo algo errado, o trabalho já está no buffer da impressora e não mais na fila, portanto não há mais o que fazer. Mas enquanto ele estiver na fila (ou no caso de ainda não ter sido enviado totalmente para a impressora), pode ser removido com o comando *lprm*. A sintaxe padrão é:

```
# lprm
```

Isso irá remover da fila o último trabalho enviado. De maneira geral, um usuário comum pode remover apenas os trabalhos criados por ele. Se você quiser remover um trabalho específico da fila, pode fazer como em uma lanchonete e “pedir pelo número”.

```
# lprm numero_do_trabalho
```

E como descobrir o número do trabalho? Com o `<C>lpq<C>`, que o mostra na coluna `<I>Job<I>`. Dependendo do tamanho do trabalho, quando descobrimos qual o número ele já foi impresso. Para resolver este problema podemos usar:

```
# lprm -a
```

Que remove **todos** os trabalhos pertencentes à mesma fila do último trabalho enviado (*nunca* faça isso como root, a não ser que tenha um ótimo motivo). Da mesma maneira que nos outros comandos, a opção *-P* pode ser utilizada para especificar uma fila em especial ou todas, com a opção *all*. O usuário root (sempre ele) pode remover trabalhos de outros usuários:

```
# lprm -Plp nome_do_usuario
```

Esse comando remove o último trabalho enviado pelo usuário *nome\_do\_usuario* à fila *lp*. Como vimos, é um comando bem simples e fácil de ser utilizado. Todos os usuários deveriam conhecê-lo, o que poderia resultar na economia de algumas toneladas de papel.

## lpc

Esse é o comando mais complexo do conjunto. Com ele controlamos todo o sistema de impressão, e podemos redirecionar trabalhos, verificar o estado das impressoras, reler os arquivos de configuração, ativar e desativar impressoras, entre vários outros truques que podem salvar nossos pescoços.

Um uso interessante é reposicionar os trabalhos na fila (útil para agradar os chefes). Tendo o número do trabalho de impressão em mãos (que pode ser conseguido com o comando *lpq*), digite:

```
# lpc topq nome_da_impresora > trabalho
```

### Quadro 1: saída do comando *lpq*

```
# lpq -Pall@maquinaremota
Rank      Owner/ID          Class  Job    Files          Size  Time
active    punk@rachael +187  A      187    /etc/fstab     690   20:35:53
done      punk@rachael +592  A      592    /tmp/a2_kB6Sgp 37002 12:31:32
```



Com isso o trabalho será passado para o início da fila. Também é possível redirecionar trabalhos de uma impressora para outra. Se uma das suas impressoras parar de funcionar e houver uma longa fila de espera, basta digitar:

```
# lpc redirect impressora_quebrada ↵
nova_impressora
```

Quando consertar sua impressora, cancele o redirecionamento trocando o parâmetro *nova\_impressora* por *off*. Se não houver outra impressora, você pode “estacionar” os trabalhos enquanto executa a manutenção:

```
# lpc holdall impressora
```

Isso irá segurar todos os trabalhos na fila. Depois de corrigir o problema, libere os trabalhos para serem impressos. Basta digitar o comando:

```
# lpc release impressora
```

É possível “segurar” e “liberar” trabalhos indicados por seus números. A sintaxe é: *lpc hold trabalho* e *lpc release trabalho*.

Outros parâmetros do LPC muito utilizados são:

- *reread*: Faz com que as configurações do servidor de impressão sejam lidas novamente. Seu uso é necessário quando os arquivos */etc/printcap* ou *lpd.conf* são alterados.
- *down impressora*: “Desliga” a impressora, fazendo com que ela pare de receber e de imprimir trabalhos
- *up impressora*: “Liga” a impressora, deixando-a pronta para receber trabalhos. Tanto neste como em vários outros comandos é possível substituir *impressora* por *all*, indicando todas as impressoras
- *status*: Mostra o estado atual das impressoras do sistema, se estão ativas, se estão redirecionadas, quantos trabalhos estão na fila etc.

Existem vários outros parâmetros que podem ser usados com o *lpc*, inclusive alguns muito úteis como o *stop*, *start*, *enable*, *disable*, *move* etc. Uma consulta à página de manual (*man lpc*) irá ajudá-lo a conhecer melhor essas opções.

Existem maneiras mais simples de configurar o LPRng (com o *apsfilter* [4], por exemplo). Mas a resolução de muitos

problemas, além de configurações mais exóticas, só pode ser conseguida com conhecimento detalhado de como as coisas funcionam, obtido através da análise dos arquivos de configuração. ■

## INFORMAÇÕES

- [1] Página oficial:  
<http://www.lprng.com>
- [2] Printing Cookbook:  
<http://www.lprng.com/PrintingCookbook/index.html>
- [3] <http://www.piterpunk.hpg.ig.com.br/>
- [4] “Apsfilter, the number one magic filter”:  
<http://www.apsfilter.org/>

## SOBRE O AUTOR

Piter PUNK, também conhecido por alguns como Roberto Freires Batista, é usuário de Linux desde 1996, e trabalhou durante vários anos como administrador de sistemas. É o principal desenvolvedor do *slackpkg*, ferramenta para atualização de pacotes no Slackware. Escrever documentação e brincar com computadores MSX são dois de seus hobbies.

