

# Recuperando a instalação do Slackware

Maio 31st, 2012 by rudsonalves

[Leave a reply »](#)

Quando se aventura por demais nas alterações do sistema, utilizando pacotes em teste, ou brincando com as configurações do *kernel*, é inevitável, uma hora ou outra algo de errado acontece. Geralmente, a maioria dos problemas podem ser evitados se forem tomados os devidos cuidados, mas há momentos de desatenção que estão além do domínio da consciência. Nestas horas o que tiver que dar errado vai dar errado.

Obviamente a diversidade de problemas possíveis é grande para tratar em um único texto e por isto vou me ater a dois problemas que me são mais comuns:

- corrigir um upgrade desastroso;
- contornar uma instalação de *kernel* problemática.

O *upgrade* desastroso a que me refiro ocorre quando o sistema fica comprometido após uma atualização mal executada. Isto geralmente acontece quando se atualiza a *glibc-solibs* com o sistema em modo 3, mais frequentemente no modo 4 (em ambiente gráfico). Sei que isto não é salutar, mas já me aconteceu algumas vezes, por negligencia e outras várias por falta de atenção (*upgradepkg \*/\*.t?z... nãooooo....*).

O segundo problema, com a instalação do *kernel*, não tem me acontecido mais tão comumente. Depois de muita surra aprendi a jamais deixar de disponibilizar uma entrada no sistema de *boot* para um *kernel* padrão, que funciona, e uma outra para o *kernel* de teste. Mas sei que isto é bem comum com outros colegas.

Em ambos os casos não é incomum o sistema trava e deixar de funcionar, mesmo após reiniciado. Muitos entram em pânico e reinstalam o sistema por completo, perdendo muito das otimizações e adaptações no processo. O texto a seguir ensina como restaurar o sistema sem ter que recorrer a uma ação extrema, com o auxílio de um *CD/DVD/Pendrive* de instalação do *Slackware*.

## 1. Corrigindo um *Upgrade* Desastroso

### 1.1. O Ato Estúpido

Não foi e primeira nem será a última vez que tenho problemas no upgrade do meu *Slackware*. O problema não é o *Slackware*, obviamente, mas o descuido do operador. O problema a tratar aqui costuma acontecer ao fazer o upgrade dos pacotes *glibc-solibs* em modo 3 ou 4, que geralmente deixa o sistema inoperante instantaneamente ou, se tiver mais sorte/azar, no próximo *boot* do sistema.

Para evitar tais erros cheguei a criar um *script* para atualizar meu sistema o qual verifica o modo de operação do sistema e não executa o upgrade caso este não seja o modo 1, mas obviamente nem sempre o uso. Mas isto não é uma tolice infundada pois nem todos os pacotes necessitam levar o sistema para o modo 1 para serem atualizados. Na verdade apenas alguns pacotes mais críticos necessitam do modo 1, os quais geralmente são disponibilizados no diretório *a/*.

Segundo o documento UPGRADE.TXT, disponível na raiz do diretório do *Slackware-current*, o pacote mais crítico no processo de atualização é o *glibc-solibs*. Na sequência, algumas outras ferramentas devem ser instaladas na ordem: *pkgtools*, *tar*, *xz* e *findutils*, todos no diretório *a/*, para que se possa ter uma atualização sem surpresas.

No entanto este texto não é sobre *upgrade* e sim sobre problemas com este. Geralmente verifico as atualizações na árvore do *Slackware-current* regularmente e, quando possível, faço as atualizações no meu sistema. A menos de alguns problemas localizados, a distribuição *current* tem me atendido



SUBSCRIBE  
Subscribe to my blogs feed



PREVIOUS ENTRY  
Correção na folha de estilo do WordPress

NEXT ENTRY  
PyQt 14 – QNetwork, baixando arquivos



#### SOBRE

Isto é uma página pessoal, onde compartilho informações, experiências, textos e aplicativos de minha autoria, ou de outros amigos e colegas da comunidade, devidamente autorizados.

Support CC - 2008

#### BLOGS

Alessandro Faria  
(CABELO)  
AlmirMendes.net  
by Ednei Pacheco!  
Leomar Viegas Jr  
LucioBig  
SlackBuilds.Org

#### META

Acessar  
Posts [RSS](#)  
[RSS](#) dos comentários  
WordPress.org

#### CATEGORIAS

Comunicados (17)  
Curtinhas (3)  
Dicas (21)  
Ferramentas (2)  
GDL64 (2)  
Hardware (6)  
Instalação (20)  
KDE4 (1)  
LibreOffice (5)  
NoFrontPage (6)  
Notícias (2)  
Packages (16)  
Programação (2)  
PyGTK (1)  
PyQt (15)  
Python (22)  
Python3 (10)  
PythonDicas (4)  
PythonTutorial (3)  
Shell Script (9)  
Slackware (18)  
Tutorial (10)  
Uncategorized (2)

#### LINUX

Alien Pastures  
Guia do Hardware.net  
PhantomX/slackbuilds  
Phoronix  
SBopkg  
SlackBlogs  
SlackBuilds dot org  
Slackware  
Slackware Doc Project  
Viva o Linux

#### NOTEBOOKS

Linux on Laptops  
TuxMobil – Linux on laptops

#### TEXTOS

Guia Desktop Linux 7.0

#### UTILITÁRIOS

Guia do Hardware.net  
Vim Bootstrap

muito bem, além de manter o sistema sempre com as últimas novidades (inclusive os últimos *bugs*). O problema ocorreu em uma destas atualizações, em que não percebi que haviam incluído uma atualização do pacote *glibc-solibs*. Quando percebi o problema já estava plantado e o sistema congelou no próximo *boot*.

## 1.2. Restaurando o sistema

Já fiz este processo de recuperação algumas vezes e não chega a ser tão trabalhoso. Necessita apenas um *CD/CDROM/pendrive* de instalação do *Slackware64/32* com *boot*. Se estiver usando o sistema *32bits* não faça isto com um *CD* com sistema *64bits*, e nem mesmo o contrário.

Reinicie o sistema e execute o *boot* normalmente como se fosse fazer outra instalação. Quando o *prompt* aparecer a primeira tarefa será limpar o sistema de arquivos de eventuais erros, devido ao desligamento forçado na última travada do sistema. Isto nem sempre é necessário, mas prefiro empregar esta medida padrão. Supondo que o seu sistema esteja no dispositivo */dev/sda*, dê os comandos abaixo:

```
root@darkstar:~# DISK=`fdisk -l /dev/sda | awk '/Linux$/ {print $1}'`
root@darkstar:~# for d in $DISK; do fsck -f $d; done
```

A primeira linha coleta todos as partições *Linux* do dispositivo */dev/sda*, a segunda linha fará a checagem do sistema de arquivos e correção dos erros. Isto pode levar alguns minutos, dependendo da velocidade do seu disco.

Feito isto monte a partição raiz do seu sistema no diretório */mnt*, para fazer a reinstalação dos pacotes com problema. No meu caso o sistema está na partição */dev/sda1*.

```
root@darkstar:~# mount /dev/sda1 /mnt
```

Agora mude para o diretório onde estão os pacotes com as atualizações que for instalar. Em meu caso, estes pacotes ficam armazenados na partição */dev/sda9*, na pasta *linux/slackware64-current/slackware64/*. Adapte as linhas a seguir às condições do seu sistema:

```
root@darkstar:~# mkdir /slackware64
root@darkstar:~# mount /dev/sda9 /slackware64
root@darkstar:~# cd /slackware64/linux/slackware64-current/slackware64/
root@darkstar:~# ls
CHECKSUMS.md5      MANIFEST.bz2    ap/    f/      kdei/   t/      xap/
CHECKSUMS.md5.asc  PACKAGES.TXT@   d/     k/      l/      tcl/    y/
FILE_LIST          a/             e/     kde/    n/      x/
```

Como pode ser visto pela saída do comando *ls*, esta é a pasta onde estão os pacotes da instalação padrão do *Slackware*.

Em seguida localize últimos pacotes atualizados para reinstalar. Isto pode ser feito verificando a listagem do diretório */var/log/packages* com o comando abaixo:

```
root@darkstar:~# ls -lat /mnt/var/log/packages
```

A opção *"t"* irá ordenando a listagem pela data de alteração/criação dos arquivos, assim basta localizar os pacotes da última atualização. Suponha que os pacotes tenham sido modificados no dia 26 de Maio, as linhas a seguir irão reinstalar os pacotes desta data:

```
root@darkstar:~# PKGS=`ls -lat /mnt/var/log/packages | \
| awk '/May 26/ {print $9}'`
root@darkstar:~# for p in $PKGS; do \
  pkg=`find . -name "$p-[0-9]*.t?z`"; \
  installpkg --root /mnt $pkg; \
done
```

Usei algumas quebras de linha para deixar o comando mais claro. A primeira linha pega a lista de pacotes instalados no dia 26 de Maio e armazena na variável *PKGS*. Dentro do *for*, a seguir, o pacote é localizado dentro dos diretórios *a*, *ap*, *d*, *e*, ... e seu caminho é passado para a variável *pkg*. A última linha reinstala o pacote sobre o sistema montado em */mnt*.

Dependendo do tipo de problema que teve na atualização, alguns pacotes ainda podem estar quebrados. Estes pacotes são marcados como *upgraded* no nome do pacote. Geralmente são pacotes que estavam sendo instalados e por algum motivo tiveram a atualização interrompida. Estes pacotes geralmente são poucos e por isto os passos seguintes serão feitos manualmente:

```
root@darkstar:~# ls /mnt/var/log/packages | grep 'upgraded'
xlockmore-5.39-x86_64-1-upgraded-2012-05-26,17:29:30
xscreensaver-5.15-x86_64-1-upgraded-2012-05-26,17:28:59

root@darkstar:~# removepkg xlockmore-5.39-x86_64-1-upgraded-2012-05-26,17:29:30
...
root@darkstar:~# removepkg xscreensaver-5.15-x86_64-1-upgraded-2012-05-26,17:28:59
...
root@darkstar:~# upgradepkg --install-new `find . -name "xlockmore-[0-9]*.t?z"`
...
root@darkstar:~# upgradepkg --install-new `find . -name "xscreensaver-[0-9]*.t?z"`
...
```

A última tarefa a fazer, antes de reiniciar o sistema, é atualizar os arquivos de configuração em */etc*. Este processo também deve ser feito manualmente, pois há arquivos de configuração que não devem ser atualizados indiscriminadamente. Isto deve ocorrer principalmente com as configurações de serviços que tenha personalizado, os arquivos de configuração do sistema como o de usuários e grupos (*passwd*, *group*, *gshadow* e *shadow*), entre outros. Use o comando *find* para localizar os arquivos em */etc*:

```
root@darkstar:~# find /mnt/etc -name *.new
/mnt/etc/rc.d/rc.M.new
/mnt/etc/group
/mnt/etc/gshadow.new
/mnt/etc/passwd
/mnt/etc/shadow.new
/mnt/etc/yp.conf.new
root@darkstar:~# mv /mnt/etc/rc.d/rc.M.new /mnt/etc/rc.d/rc.M
root@darkstar:~# rm /mnt/etc/yp.conf.new
root@darkstar:~# diff /mnt/etc/shadow.new /mnt/etc/shadow
...
```

Este processo é bem pessoal. No exemplo acima movi o */mnt/rc.d/rc.M.new* para */mnt/rc.d/rc.M*, supondo que ele não tenha sido editado. O *yp.conf.new* se refere ao serviço *NIS* (*Network Information Server*), que estou supondo que não tenha sido modificado. Os arquivos *shadow.new*, *gshadow.new* em geral devem ser removidos, no entanto é uma boa prática compará-los com as versões anteriores para verificar se não existe algum usuário ou grupo padrão (novos), que o sistema atualizado possa vir a necessitar.

Após tudo isto o sistema está pronto para reiniciar.

## 2. Contornar uma Instalação de *kernel* Problemática

Geralmente o problema com a personalização do *kernel* está na inclusão do controlador do disco adequado para o seu hardware (*Device Drivers*->*Device Drivers*) e/ou a seleção do sistema de arquivos (*File systems*) necessário para montar a partição raiz. Estes pontos são críticos pois sem eles o sistema não reconhece o seu disco ou não consegue montar o sistema de arquivos nas partições. Estes itens devem ser incorporadas ao *kernel*, uma vez que necessitam ser carregadas a toda inicialização. Não faz sentido deixá-las como módulo em um *kernel* personalizado. No demais você pode ficar sem som, *Bluetooth*, *usb*, *Wireless* entre outros dispositivos, mas geralmente o sistema consegue iniciar.

Caso o sistema trave e por algum motivo estranho não tenha outras entradas de reparo em seu *lilo.conf* ou *grub.cfg*, entre com um *CD/DVD/Pendrive* com a instalação do *Slackware* e reinicie o sistema. Use a linha de comando a seguir, no *prompt* do boot:

```
huge.s root=/dev/sda1 rdinit= ro
```

Isto diz ao sistema de *boot* para carregar o *kernel huge.s* e montar a raiz do sistema em */dev/sda1*, como *readonly*. O *rdinit* pode ser útil caso o tenha gerado um *ramdisk* inicial para o *kernel* do

CD/DVD/Pendrive que está *bootando*, se não for o caso pode deixar isto em branco como está. Substitua o `/dev/sda1` pela partição em que o seu sistema foi instalado.

Observe que esta linha aparece na mensagem de *boot* do disco de instalação, quatro linhas acima do *prompt*, exatamente como a escrevi.

```
LIOLINUX 3.04 2009-12-10 ETCB Copyright (C) 1994-2009 H. Peter Anvin et al
Welcome to Slackware64 version 13.37 -current (Linux kernel 2.6.30.7)

If you need to pass extra parameters to the kernel, enter them at the prompt
below after the name of the kernel to boot (huge.s etc).

In a pinch, you can boot your system from here with a command like:
boot! huge.s root=/dev/sda1 rdinit=rs

In the example above, /dev/sda1 is the / Linux partition.

To test your memory with memtest86+, enter memtest on the boot line below.

This prompt is just for entering extra parameters. If you don't need to enter
any parameters, hit ENTER to boot the default kernel "huge.s" or press 1/2/
for a listing of more kernel choices.

boot! _
```

Após isto pressione enter e deixe o sistema iniciar. Quanto o sistema estiver iniciado edite o `/etc/lilo.conf`, ou o `/boot/grub/grub.cfg`, e corrija a entrada para um *kernel* funcional.

Se ainda for fazer mais testes com a compilação do *kernel*, considere adicionar uma entrada de Teste no seu `lilo.conf`, ou o `grub.cfg`.

## <2>Considerações Finais

Não escrevi muito sobre a restauração do *kernel*, por ser um problema mais simples, mas reconheço que existem outros detalhes a serem adicionados ali. Talvez faça isto em outro texto, caso achar necessário mais explicações.

Espero que o texto seja útil de alguma forma, mas a melhor forma de agir é prevenir tais operações desastrosas. Como não sou administrador de sistema, não tenho grandes problemas em um pouco de caos em minhas máquinas.

- [Previous Entry: Correção na folha de estilo do WordPress](#)
- [Next Entry: PyQt 14 – QNetwork, baixando arquivos](#)

Posted in [Instalação](#), [Slackware](#)

Tags: [bash](#) [Slackware](#) [upgrade](#)

You can follow any responses to this entry through the [RSS 2.0 Feed](#). You can [leave a response](#), or [trackback](#) from your own site.

## Advertisement

# 2 comments

[Add your comment](#)



Decio

10 de Maio de 2017 às 2:09 PM

Olá, estou com o erro ao qual você se refere na primeira parte deste tuto. Enquanto fazia o processo me deparei com erro neste comando:  
PKGS='ls -lat /mnt/var/log/packages \  
| awk 'May 26/ {print \$9}'  
/bin/ls: cannot access ' ': No such file or directory

Responder



[rudsonalves](#)

1 de novembro de 2017 às 6:44 PM

Desculpe Decio, não vi sua mensagem.

Mas se for útil a alguém, observe que o texto pressupõe que o sistema está montado na partição `/mnt`. Outra possibilidade é que o disco tenha sido corrompido e por isto não acesse ou não exista o diretório `/var/log/packages` em `/dev/sda1`.

Responder

## Deixe uma resposta

Nome (required)

Mail (will not be published) (required)

Site

Submit Comment