



Utilizando o



archlinux™
A simple, lightweight linux distribution.

no Desktop

UTILIZANDO O ARCHLINUX NO DESKTOP

Sérgio Berlotto

Dedicatória

Dedico este livro e agradeço a minha esposa e ao meu filho, por me darem sempre alegrias e serem a base de muitos aprendizados, e a comunidade Open Source, principalmente ao Archlinux-BR, que me acolheu e me ajudou sempre quando necessário.

Índice

| | |
|--|-----|
| Prefácio | iii |
| 1. Introdução | 1 |
| 1.1. Conhecendo o Arch Linux | 1 |
| 1.2. Instalação e Configuração | 5 |
| 2. Gerenciamento de Pacotes | 10 |
| 2.1. Pacman | 10 |
| 2.2. Interfaces Gráficas para o Pacman | 10 |
| 2.3. Repositórios | 12 |
| 2.4. Como criar um pacote para o Arch Linux | 15 |
| 2.5. Como enviar pacotes para o AUR | 21 |
| 3. ABS/AUR | 23 |
| 3.1. Arch Build System (ABS) | 23 |
| 3.2. ArchLinux User Repository (AUR) | 23 |
| 3.3. Trusted Users | 25 |
| 3.4. Compilando através do ABS | 25 |
| 4. Mais informações e contribuições | 28 |
| 4.1. Projetos Relacionados | 28 |
| 5. Perguntas Frequentes | 30 |
| 6. Listagem de comandos úteis de modo geral para o ArchLinux | 32 |
| 7. Instalação e configuração de programas | 34 |
| 7.1. Como instalar ambiente gráfico base | 34 |
| 7.2. Como instalar um ambiente GNOME | 36 |
| 7.3. Como instalar um ambiente KDE 4 | 37 |
| 7.4. Como instalar um ambiente XFCE | 37 |
| 7.5. Como instalar o SOM | 38 |
| 8. Como compilar o kernel do Jeito Arch ¹ | 40 |
| 8.1. Kernels diferenciados | 40 |
| 8.2. Dicas de Performance | 41 |
| Considerações Finais | 43 |

Prefácio

Conheça tudo sobre o ArchLinux!

Este livro é destinado a pessoas que gostem de se aventurar em novos mundos e que gostem principalmente de Software Livre.

Neste livro, eu irei lhes mostrar e falar bastante sobre a distribuição linux conhecida como Arch Linux. Você, leitor, conhecerá desde a história inicial da distribuição, todas as suas características e artimanhas de utilização, desde a instalação básica até a instalação de programas, ambientes de trabalho, kernels diferenciados, compilação de código e geração de pacotes.

Lendo este livro até o fim, com certeza você estará apto a instalar, manter e utilizar no dia-a-dia uma distribuição Arch Linux.

Licença:



Esta obra está licenciada sob "Creative Atribuição-Compartilhamento pela mesma Licença 3.0 Brasil Commons". Para mais detalhes visite o site <http://creativecommons.org/licenses/by-sa/3.0/br/>.

Versão do Livro:

1.0

Ferramentas:

Este livro foi escrito utilizando as ferramentas: asciidoc, gedit e Arch Linux.

Linguagem: Português do Brasil (pt_BR)

Site Oficial do Livro:

No site <http://archlinuxbook.wordpress.com>, você pode encontrar a versão mais atualizada e maiores informações sobre esta obra.

Data: Novembro / 2010

Sobre o Autor:

Sérgio H. Berlotto Jr, gaúcho, nasceu em Janeiro de 1981, é desenvolvedor desde 1999, e *linuxer* deste meados de 2003. A partir de 2009, conheceu o ArchLinux, que, desde então, tornou-se sua distribuição oficial e de uso diário, com a qual adquiriu muito conhecimento sobre este ambiente e criou um apego pela filosofia da distribuição.

Site: <http://berlotto.blog.br>

E-Mail: sergio@archlinux.com.br [<mailto:sergio@archlinux.com.br>]

Capítulo 1. Introdução

Este livro foi criado para servir de base a todos aqueles que queiram utilizar a distribuição Arch Linux ¹ em seu computador de casa ou do escritório. Eu falo, explico e mostro vários tópicos e vários aspectos desta distribuição, explicando os caminhos mais fáceis e/ou interessantes para a instalação e utilização desta distro que vem crescendo dia a dia.

Este livro é algo como um manual misturado com um guia de usuário!

Neste livro mostraremos também alguns pontos importantes do Arch Linux, arquivos de configuração, como manter a estabilidade, alguns drivers de vídeo, som e gerenciador de janelas, que é o mínimo necessário para você ter um ambiente desktop inicial para trabalho e diversão.

Este livro é voltado aos usuários do Brasil (pt_BR), porém pode servir como base para qualquer usuário instalar e manter o ArchLinux em qualquer outra linguagem também, alterando poucos itens.

Depois de ler este livro, você estará apto a manter corretamente sua distribuição Arch Linux funcionando, com seus programas mais utilizados, com estabilidade e sabendo também onde buscar as informações necessárias para as adversidades do dia-a-dia, que por sinal, não são muitas, mas existem !

1.1. Conhecendo o Arch Linux

1.1.1. História e Conceito

O Arch Linux é uma distribuição GNU/Linux ² que foi criada em 2002, por Judd Vinet, que em 2007 repassou a liderança do projeto a Aaron Griffin.

Ela é uma distribuição minimalista, criada do zero, concebida a partir das ideias do antigo Crux, porém com um robusto gerenciamento de pacotes e resolução de dependências entre eles (pacman e makepkg).

O conceito do Arch é o velho conhecido KISS (Keep it Simple Stupid), ou seja, mantenha tudo o mais simples possível.

A distribuição mantém a cultura minimalista e simplificada. Com a sua instalação "não vem nada", ou seja, quando baixamos a imagem ISO para instalá-lo, instalamos somente o sistema básico, com a velha conhecida linha de comando. E a partir daí é que colocamos a mão na massa para customizar nossa instalação conforme queremos. Mas não se assuste! Não temos janelas bonitinhas de configuração e wizards para instalações, porém temos a simplicidade nos arquivos de configuração e algumas interfaces de auxílio diretas, e práticas, via terminal. Quem tem medo da "tela preta" do Linux não deve se aventurar por aqui, mas nunca é tarde para aprender! Por isto, sempre se enfatiza que: O Arch Linux é uma distribuição para usuários avançados e não para iniciantes!

Muitos usuários de linux comparam o Arch com o Slack, pois os dois têm a filosofia da simplicidade, porém a grande diferença (pelo menos a mais comentada) que existe entre as duas distros é o

¹Site Oficial do projeto ArchLinux [<http://www.archlinux.org>]

²GNU é o nome do projeto e propriedade da Free Software Foundation, e Linux é nome do kernel do sistema, criado por Linus Torvalds

gerenciamento de pacotes do Arch. Que nos trás muita organização e menos trabalho para criar, gerenciar e instalar os pacotes no ambiente.

"The Arch Way"

Como sabemos, e vemos escrito em vários sites, e, vários usuários comentam também é que: o Arch Linux segue a linha da simplicidade. Mas como isto ? Vamos ver abaixo alguns itens que estão descritos no artigo "The Arch Way", encontrado no próprio site internacional da distro, e aí você vai entender melhor tudo que está sendo falado, e poderá, com mais sabedoria, decidir se esta é a melhor distribuição para você!

Simplicidade

O Arch é completamente voltado a simplicidade. Mas esta afirmação pode te levar a vários pontos de vista, então entenda: simplicidade no Arch é definida como "sem acréscimos desnecessários, modificações ou complicações".

No Arch, o usuário pode moldar o sistema completamente de acordo com suas necessidades, ficando assim com a sua cara, com seu jeito. A simplicidade do Arch traz consigo muitas facilidades, que se tornam o grande trunfo da distribuição, e você consegue ver isto logo de cara, com pouco tempo de uso.

O Arch mantém um sistema GNU/Linux com suas facilidades e dificuldades da mesma forma em que são encontradas, porém organizadas e transparentes. Esconder as complexidades de um sistema resulta geralmente em um sistema muito mais complexo, e isto é evitado. Resumindo: o Arch é uma distribuição elegante e minimalista!

Correção de Código sobre conveniência

O Arch Linux precede sobre a elegância do design, bem como a clareza, código simples, ao invés de arranjos desnecessários, automação, eye candy ou "newbie-friendliness". Os patches do software são mantidos a um mínimo absoluto — de preferência, nunca. A implementação simples é sempre um trunfo da interface de usuário simples.

Simplicidade de implementação, elegância de código, e minimalismo serão sempre as prioridades regentes do desenvolvimento do Arch.

Conceitos, designs e funcionalidades são geradas e implementadas pelo uso dos princípios do Arch Way como um guia, ao invés de influências externas. A equipe de desenvolvimento é decidida no seu empenho e dedicação a filosofia do Arch Way. Se você compartilha essa visão, você é bem-vindo e incentivado a usar o Arch.

Centrado no Usuário

Diferentemente de muitas distros que são "*user-friendly*", ou seja, amigável, confortável ao usuário, "facilitando" a utilização do sistema com muitas features e alterações, o Arch é "*user-centric*", é direcionado ao usuário. O usuário do arch é o centro das atenções, ele é quem manda. É um ponto de vista diferente, apesar de soar levemente parecido.

O Arch direciona-se a, e, acomoda competentes usuários de GNU/Linux dando-lhes o controle completo e a total responsabilidade sobre o sistema. E este modo *user-centric* de ser, implica em obrigatoriamente em um modo *do-it-yourself* (faça você mesmo) de ser do usuário. Ao invés de solicitar um assistência paga, ou solicitar novas features aos desenvolvedores, você é induzido a criar, e a resolver a maioria dos problemas sozinho ou com a ajuda da comunidade, que tem ótima documentação e canais de comunicação, como o fórum, wiki e lista de discussão.

Tanto isto é verdade que o Arch tem uma espécie de repositório especial para guardar os pacotes gerados pelos usuários normais, que são aqueles que não são desenvolvedores oficiais nem *Trusted Users*. Isto ajuda muito no desenvolvimento da distribuição, pois como diz o ditado: *2 cabeças pensam mais do que 1!* E assim, várias ideias e inovações em programas e bibliotecas são adicionadas aos repositórios oficiais vindas das várias contribuições disponibilizadas.

Abertura

A abertura vai de mãos dadas com a simplicidade, e é também um dos princípios orientadores do desenvolvimento do Arch Linux.

O Arch Linux utiliza ferramentas simples, que são selecionadas ou construídas com a abertura dos fontes e sua saída em mente.

A natureza aberta do Arch Linux também implica numa curva de aprendizagem bastante acentuada, mas os usuários experientes tendem a encontrar outros sistemas fechados mais inconvenientes para controlar. O princípio da abertura estende-se aos membros da comunidade. Os usuários do Arch Linux são conhecidos por serem abertos a ajudar e aconselhar, bem como com contribuições de pacotes para a comunidade mantenedora do Arch User Repository (AUR).

Liberdade

A liberdade que se fala neste ponto, não é somente a liberdade de escolha de pacotes, mas também a liberdade que o usuário tem na hora de "formar" seu sistema. O Arch do usuário pode ser o que ele quiser, seja qual for a direção escolhida. Você nunca será obrigado a nada utilizando o Arch. Tudo é feito na maior transparência e simplicidade. O usuário pode escolher em fazer tudo na mão, escolher entre pacotes livres e/ou proprietários, pode definir quais softwares rodarão no sistema, quais serviços estarão no ar e quais não estarão. É o usuário quem escolhe !

Ao manter o sistema simples, o Arch Linux fornece a liberdade para fazer qualquer escolha sobre o sistema.

O ArchLinux é o que você faz!

— Criador do projeto ArchLinux *Judd Vinet*

1.1.2. Inicialização

O Arch Linux utiliza um sistema de inicialização identificado como "BSD Like Init Scripts", diferentemente de várias outras distros que temos disponíveis. Mas o que significa isto ? Qual o sistema de inicialização que tem na outros distros ? Como funciona ? Tentarei explicar como funcionam da melhor forma possível, para que você possa conhecer e ter, daqui para frente, alguma ideia de como dar manutenção em um ambiente destes. Então, vamos lá!

Todos sabemos que os init scripts, em um sistema unix like servem para inicializar o sistema de fato, ou seja, chamam os scripts e daemons de cada serviço conforme estão configurados, e são executados logo que o boot da bios acontece.

Ambos, System V e BSD-Like fazem isto, mas qual a diferença entre os dois sistemas ? A diferença entre os dois sistemas de inicialização está na sua estrutura.

System V

O System V tem uma estrutura baseada em runlevels. Ele executa os daemons conforme cada runlevel é atingido no sistema. Geralmente temos de 6 a 7 runlevels na maioria dos sistemas unix-like, e não existe obrigatoriedade na sua utilização.

Por exemplo, em um sistema Debian temos a configuração do init no arquivo */etc/inittab*. Nos diretórios */etc/init.d/rcN.d* ficam localizados links simbólicos para os scripts que estão fisicamente gravados em */etc/init.d/*, onde este "N" do "rcN.d" indica o número do runlevel em que o script ou daemon será executado. E, nesta mesma organização de um Debian Linux, temos 7 runlevels, conforme a lista abaixo:

- 0 = pára o sistema
- 1 = modo monousuário
- 2 a 5 = modos multiusuário
- 6 = reinicialização do sistema

Este é um ótimo exemplo de um sistema que utiliza o modo System V, e entre as outras distribuições que também utilizam o System V como modo de inicialização, (por exemplo Red Hat e Suse) podemos ter pequenas alterações, principalmente na definição dos runlevels, para que se adaptem a cada situação.

BSD Like

Já o modo BSD de inicialização, tem como característica a simplificação das configurações, onde todo o processo é baseado em um arquivo, o */etc/rc.conf*, e os scripts estão em apenas um diretório, o */etc/rc.d*, sendo que não temos a utilização dos runlevels neste caso, mas sim apenas dois modos de trabalho: Monousuário ou Multiusuário. Então, se você quer inicializar algum serviço no sistema, basta adicioná-lo no arquivo */etc/rc.conf*, ou ainda em alguns sistemas BSDs, no arquivo */etc/rc.conf.local*.

Dois grandes exemplos de como um linux com sistema de inicialização BSD Like funciona são: ArchLinux e Slakware. No Arch, é exatamente como mencionado acima. A inicialização do sistema é baseada no arquivo */etc/rc.conf*. Além dos daemons que inicializarão ao iniciar o sistema, podemos também configurar relógio, linguagem, teclado, lvm, hostname e rede. Todos os scripts de daemons (ou serviços) estão localizados em */etc/rc.d/* e podem ser chamados manualmente com os parâmetros "start" e "stop", além de outros específicos de cada serviço, conforme a nossa necessidade.

Alguns desenvolvedores (e utilizadores também) acham que o System V é mais complicado de configurar, porém muito flexível, e que o BSD Like é bastante simples, e de fácil aprendizado, mas bem menos flexível.

1.2. Instalação e Configuração

1.2.1. Instalando o ArchLinux

Agora que temos já algum conhecimento sobre como funciona e o que é o ArchLinux, vamos colocar a mão na massa e instalá-lo. Uma dica minha é: caso você não esteja muito familiarizado com o ambiente do Arch e quer testar qualquer coisa, instale-o em uma Virtual Machine, assim você pode fazer e perder e refazer quantas vezes for necessário.

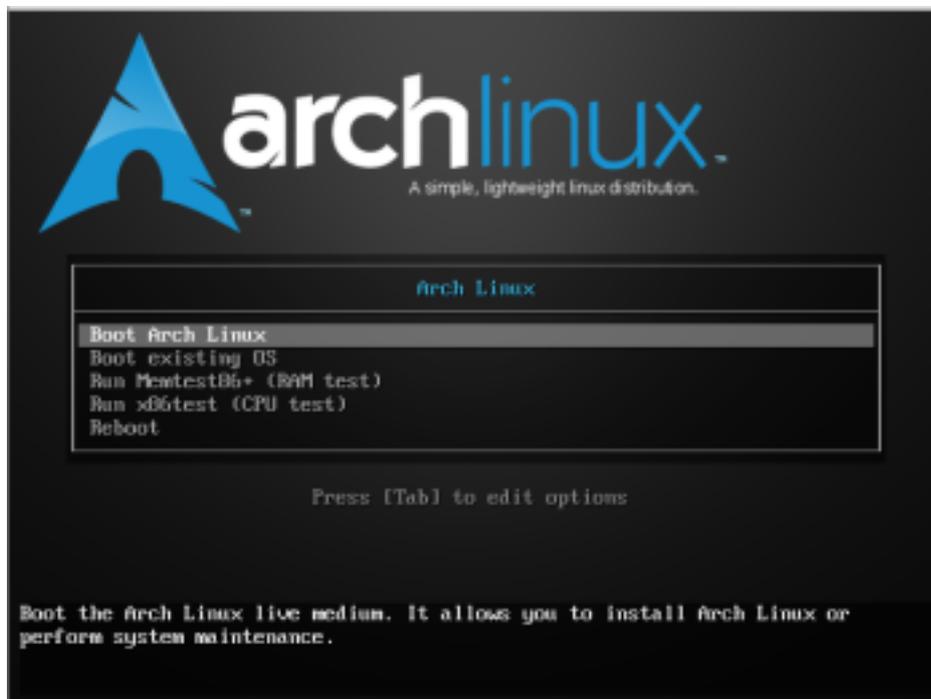
A instalação começa com o download da ISO.

Acesse: <http://www.archlinux.org/download> e baixe a iso que diz "netinstall", que traz somente o necessário para rodar o ambiente de instalação, em mais ou menos 160MB. Neste ponto tem dois detalhes importantes:

1. Arquitetura: você deve baixar a iso que traz no nome do arquivo o nome da sua arquitetura: i686 ou x86_64.
2. Imagem Híbrida: Tem outra ISO que contém uma imagem híbrida do Arch, ou seja, ela contém ambas arquiteturas e pode ser gravada tanto em CDs como em pendrives, assim nos facilitando muito a vida quando queremos principalmente instalar várias máquinas.

Bom, vamos então a nossa instalação!

Esta é a tela de inicialização da instalação do Arch.



1. Escolha "**Boot Arch Linux**"
2. Deixa carregar o sistema e insira o usuário "root", sem senha.
3. Execute "km", que é configurador do teclado, escolha "i386/qwerty/br-abnt2.map.gz", e na segunda opção pode clicar em "skip".

4. Agora iniciaremos a instalação, execute `/arch/setup`
5. Clique "Ok" na tela de boas vindas.
6. Os menus desta instalação devem ser executados todos praticamente, e na ordem em que apareçam na tela.
 - a. **Select Source**
 - i. Selecione a fonte dos arquivos de instalação, e se você escolher "net" ele solicitará que você configure a rede. DHCP Geralmente resolve nosso problema. Ou então você pode também configurar sua rede de IP Fixo.
 - ii. Agora, selecione o mirror para baixar os arquivos, pode ser qualquer um que termine com ".br"
 - b. **Set Clock**
 - i. Selecione a sua região "Americas#Sao_Paulo".
 - ii. Sete a hora, ou diga que vai ser atualizado pela hora UTC (melhor opção)
 - c. **Prepare Hard Drives** (Particionamento)
 - i. Podemos escolher por particionar automaticamente, onde ele decide o que é melhor e nos pergunta somente alguns tamanhos.
 - ii. Ou podemos escolher particionar manualmente e depois configurar os pontos de montagem, onde define-se onde está o "/", o "/home", o "/boot", "swap", etc...
 - d. **Select Packages**
 - i. Aqui ele mostrará alguns pacotes que estão disponíveis, e muitos já marcados. Eu aconselho a não desmarcar nenhum caso você não saiba exatamente o que está fazendo, e marcar pelo menos o pacote "sudo" a mais, pois vai utilizá-lo em vários momentos. *Dica kiss*: Não marque pacotes que você **acha** que pode utilizar mais tarde! Isto acaba instalando libs e programas desnecessários, e não queremos isto.
 - e. **Install packages**
 - i. Aqui serão verificados e instalados os pacotes escolhidos no passo anterior. Aguarde terminar...
 - f. **Configure System** Este é o passo mais importante, mas nem por isto o mais difícil! Veja abaixo os itens que devem, no mínimo, ser verificados, pois nem todos necessitam de alteração, pelo menos não neste momento, e os que não foram citados na lista abaixo, podem, caso necessário, ser editados depois, em casos mais especiais.
 - i. `/etc/rc.conf` → Este é o arquivo de configuração do sistema, o mais importante. Veja os itens que necessitam ser verificados e alterados se necessário, devem ficar com valores aqui citados/explicados:
 - `LOCALE=pt_BR.utf8`

- *HARDWARECLOCK*= → Este item deve estar correto conforme configurado anteriormente
 - *TIMEZONE*= → Este item deve estar correto conforme configurado anteriormente
 - *KEYMAP*="br-abnt2"
 - *HOSTNAME*=NomeDaMaquina → Escolha um nome para seu host e coloque-o aqui
 - *eth0*="dhcp" → Informações sobre a rede. O mais fácil é deixar como "dhcp", isto se você está em uma rede que contenha um servidor DHCP, mas você pode definir seu ip fixo também, como por exemplo "eth0="eth0 192.168.0.2 netmask 255.255.255.0 broadcast 192.168.255.255"".
 - *lo*="localhost 127.0.0.1" → este item deve ser acrescentado
 - *INTERFACES*(*lo eth0*) → estas são as configurações de rede, definidas acima, que estarão disponíveis e serão conectadas.
 - *DAEMONS* → A lista dos serviços que serão iniciados com o sistema. Não coloque nada agora, somente depois conforme for sendo necessário ao utilizar o sistema.
- ii. */etc/fstab* : Confira aqui os pontos de montagem do sistema, e pode acrescentar alguma outra, caso queira.
- iii. */etc/resolv.conf* : Ajuste os IPs dos servidores de DNS que serão utilizados por você. Pode colocar aqui os do Google ou do OpenDNS por exemplo.
- iv. */etc/hosts* : É neste arquivo que você define os nomes de máquinas na rede e seus respectivos Ips. Acrescente as que necessitar, e se necessitar.
- v. */etc/locale.gen* : Edite este e descomente aqueles que forem iguais ao LOCALE informado no */etc/rc.conf*, no caso "pt_BR.UTF8"
- vi. */etc/pacman.conf* : A configuração do nosso gerenciador de pacotes, o PACMAN, descomente as linhas "ShowSize" e "TotalDownload".
- vii. */etc/pacman.d/mirrorlist* : Configurar os repositórios que devem estar já habilitados: core, extra e community . Aqui você escolhe quais mirrors de rede estarão disponíveis para buscar os pacotes. Descomente os brasileiros, que tem ".br" no nome por exemplo. É aconselhado que não utilize o archlinux.org por estar geralmente com muito tráfego.
- g. Install Bootloader**
- i. Aqui vamos instalar o grub, você será questionado por onde deve instalar o bootloader.
- h. Exit Install**
- i. Nem preciso falar nada sobre este passo. Ok ! Saída e depois disto ele indica a você para executar um reboot para iniciar no novo sistema!

Parabéns ! Você é um novo ArchLinux User!!!

1.2.2. Pós instalação

Agora, faremos alguns ajustes para o correto/melhor funcionamento da maioria das coisas que mais se utiliza no nosso ambiente:

Um detalhe, esta seção é uma lista de tarefas e dicas do que fazer depois da instalação, para guiar o usuário. Tarefas mais simples conterão seus comandos aqui mesmo, listados no livro, e tarefas mais complicadas, que não estiverem contidas neste livro, estarão seguidas pelos seus links do wiki. Ok !

- Atualizar os repositórios e o sistema:

Primeiro teste sua conexão

```
# ping -c 3 www.google.com
```

```
# pacman -Syu
```

- **Pacman colorido**

Com estes comandos, os resultados do pacman ficam mais agradáveis no terminal

```
# pacman -S pacman-color
```

Editar o arquivo `~/.bashrc` e acrescentar:

```
alias pacman="pacman-color"
```

- **Execute a verificação dos mirrors para melhor performance do pacman**

```
cd /etc/pacman.d
cp mirrorlist mirrorlist.backup
//Edite o mirrorlist.backup e descomente todos os mirrors que são
//mais próximos geograficamente a você
nano mirrorlist.backup
rankmirrors -n 6 mirrorlist.backup > mirrorlist
pacman -Syy
```

- **Crie do seu Usuário**

```
adduser meulogin
```



Dica de Segurança

Sempre crie um usuário comum para você utilizar, é importante isto! **NUNCA UTILIZE O SISTEMA, DIA-A-DIA, COM UM USUÁRIO ADMINISTRADOR, EM NENHUM SISTEMA OPERACIONAL!**

- **Colocar o usuário em vários grupos pertinentes**

```
# gpasswd -a seulogin audio,lp,optical,storage,network
# gpasswd -a seulogin video,wheel,games,power,scanner
```

- **Instale o yaourt** : ferramenta para trabalhar com os pacotes ao AUR mais facilmente (item Seção 3.2.1, “Yaourt”)

- **Instale o driver de Vídeo correto** (itens Seção 7.1.1, “Instalação do driver NVidia” e Seção 7.1.2, “Instalação do driver ATI Catalyst”)
- **Instale seu ambiente gráfico predileto** (itens Seção 7.2, “Como instalar um ambiente GNOME”, Seção 7.3, “Como instalar um ambiente KDE 4” e Seção 7.4, “Como instalar um ambiente XFCE”)
- **Instale mais algumas fontes**

```
pacman -S xorg ttf-dejavu ttf-bitstream-vera ttf-ms-fonts
```

- **Acesso a câmeras digitais**

Instalar o libgphoto

```
# pacman -S libgphoto2
```

Adicionar o usuário no grupo *camera*:

```
# gpasswd -a seulogin camera
```

- **Instale um ambiente gráfico mais aprimorado**, caso goste. Link: <https://wiki.archlinux.org/index.php/Compiz>
- **Instale codecs de mídia**, para tocar música e vídeos. Link: <https://wiki.archlinux.org/index.php/Codecs>
- **Instale o browser que mais gosta e os seus plugins para Flash e Java**. Links:
 - <https://wiki.archlinux.org/index.php/Chrome>
 - <https://wiki.archlinux.org/index.php/Firefox>
 - https://wiki.archlinux.org/index.php/Browser_Plugins
- **Instale as libs necessárias para rodar DVDs**. Link: <https://wiki.archlinux.org/index.php/DVD>
- **Instale os (des)compactadores:**

```
# pacman -S tar unrar unace unzip zip bzip2 gzip p7zip
```

- **Instale o Acrobat Reader** para ver seus PDFs (Este está presente no repositório *archlinuxfr*)

```
# pacman -S acroread
```

- **Instale agora outros programas** que, mesmo que não tenham sido citados aqui podem ser interessantes para você, ou deixe isto para depois e vá instalando conforme a necessidade de uso.

Segue o link para uma lista dos programas mais comuns: https://wiki.archlinux.org/index.php/Common_Applications

Capítulo 2. Gerenciamento de Pacotes

2.1. Pacman

Assim como várias distribuições, o Arch tem seu próprio sistema de pacotes, e o gerenciador é chamado de "pacman". Simples, fácil e muito funcional! Aqui, tem um pequeno, mas importante, detalhe: todos os pacotes do Arch são compilados para processadores a partir de i686 (32 e 64 bits), ou seja, processadores antigos não funcionarão com esta distro.

O pacman está na sua versão 3.4.0, lançada em Junho de 2010.

Ele gerencia os pacotes em formato binário, ou seja, os executáveis já compilados, de forma inteligente e fácil de se utilizar, e estes binários são gerados com ABS (Arch Built System). O pacman funciona baseado em um arquivo de configurações, que contém além de algumas diretivas para seu funcionamento, a lista de repositórios habilitados, que informam onde estão hospedados os pacotes que serão utilizados no sistema.

O pacman tem uma forma padrão de funcionamento, porém encontramos algumas maneiras de melhorar sua performance.

Uma das características do pacman é que ele guarda suas informações em uma base de dados local para seu controle. E com o comando `pacman-optimize && sync` você consegue otimizar esta base, fazendo com que ele a acesse de forma otimizada e mais ágil.

Outra ferramenta interessante é o pacote: `pactools`. Ele contém alguns programinhas interessantes, um deles é o `pt-pacman-cage`, que faz com que o arquivo da base de dados local esteja localizado em um local no HD que seja contínuo, diminuindo assim a tempo de acesso a este.

Outra forma ainda de tentar aumentar a performance do pacman, é melhorar a velocidade download dos pacotes. No arquivo de configuração (`/etc/pacman.conf`) temos a propriedade "XferCommand" onde podemos alterá-la para utilizar diferentes ferramentas para efetuar o download de cada pacote. Podemos configurar por exemplo o `aria2c`, o `wget`, o `script` `pacget`, etc. E tem também o `powerpill`, que é um programa paralelo ao pacman, que utiliza o `aria2c` para fazer os downloads, aumentando a performance dos mesmos.

E com o programa "rankmirror", teremos uma organização dos repositórios, que serão organizados conforme a melhor avaliação da velocidade de download. Assim ajudando a melhorar ainda mais o download.

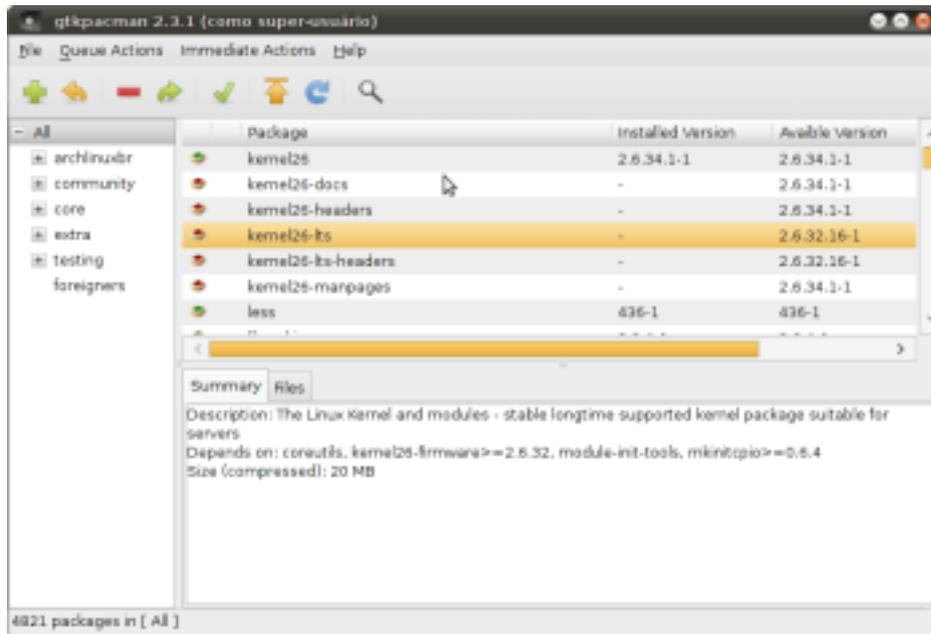
2.2. Interfaces Gráficas para o Pacman

O pacman conta com algumas interfaces gráficas, também. Elas originalmente não fazem parte do pacote padrão da ferramenta, mas ajudam a quem gosta de trabalhar com elas.

Eis alguns exemplos:

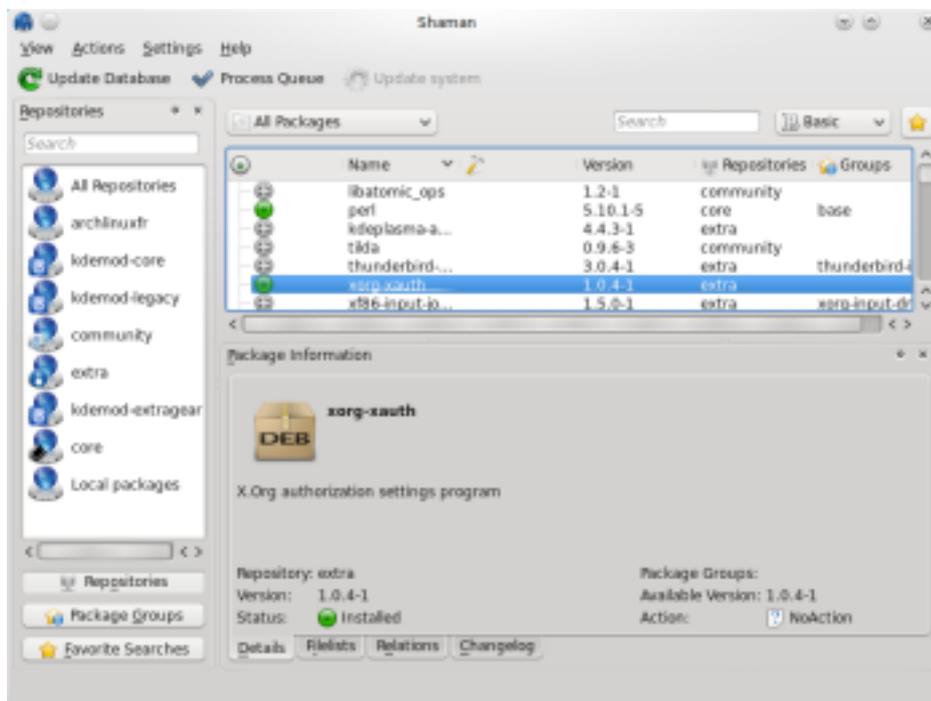
GtkPacman

O GtKpacman é bem simples mas eficiente, faz procuras de pacotes, mostra informações de pacotes e listagem dos repositórios também. Além de, claro, instalar, remover e atualizar pacotes.



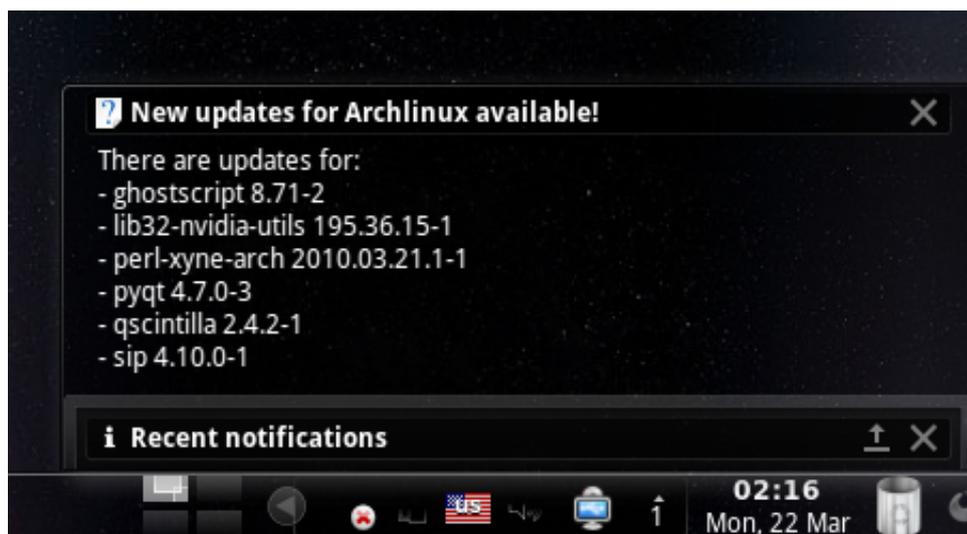
Shaman

O Shaman é uma ferramenta que tem a mesma função do GtKpacman, porém é parte do projeto Chakra e utiliza a biblioteca Qt e, além disto, é integrada ao ABS, buscando pacotes do AUR também.



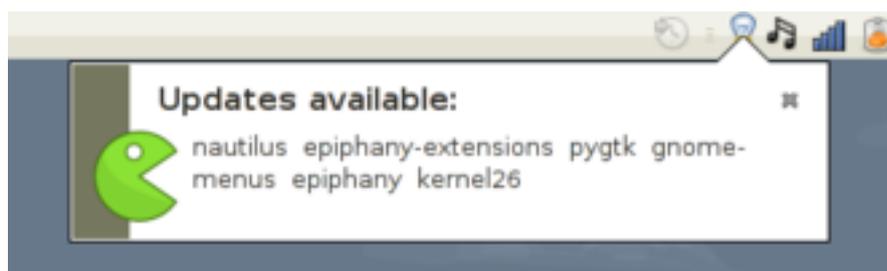
Archup

O Archup é um pequeno aplicativo de verificação de atualizações de pacotes para o Arch. Ele é criado em C utilizando a LibNotify. Roda tanto para o KDE quando para o Gnome. <http://www.nongnu.org/archup/>



Alunn

O Alunn é outro programa que simplesmente fica verificando o status dos pacotes no seu sistema e avisa quando tem atualizações disponíveis, com uma mensagem na bandeja do sistema.



E além destas citadas tem outras também que efetuam tarefas parecidas com algumas variações de biblioteca gráfica e organização. Você pode saber mais no wiki oficial https://wiki.archlinux.org/index.php/Pacman_GUI_Frontends .

2.3. Repositórios

Geralmente estamos acostumados a ver sempre um formato de repositórios parecidos com o do Debian, onde temos o repositório da versão, e dentro dele a separação de stable, com os pacotes estáveis, unstable com os pacotes ainda instáveis, testing para os pacotes que estão sendo testados, e outros as vezes. Porém no Arch é diferente. Já que não temos uma versão da distro, por ser "Rolling Release", todos os pacotes funcionam em todas as instalações, ou seja, não temos a separação de versão. Assim, um pacote do gnome 2.30, por exemplo, serve tanto para uma nova instalação do Arch, bem como para uma instalação feita a muito tempo atrás.

Os repositórios no Arch estão separados na seguinte forma:

core

Pacotes principais da distribuição e alguns opcionais

extra

Pacotes extras da distribuição que não estão no core

unstable

Pacotes experimentais e instáveis

testing

Pacotes que são candidatos aos repositórios core, extra e unstable

release

Pacotes históricos, não é para uso regular

community

Pacotes que do AUR que são mantidos pelos TUs e tem votos suficientes para entrar neste repositório

unsupported

Este não contem pacotes binários, mas PKGBUILDS que são submetidos por usuários regulares

Tem também alguns repositórios não oficiais que são mantidos por alguns usuários, empresas e/ou grupos de usuários que contém os mais diversos pacotes, como por exemplo :

archlinuxbr

Repositorio Brasileiro

archlinuxfr

Repositório Francês

archgames

Repositório de jogos

entre outros...

Existe uma lista completa de repositórios não oficiais que pode ser encontrada em: http://wiki.archlinux.org/index.php/Unofficial_User_Repositories .

2.3.1. Servidores Espelho (Mirror)

Para cada repositório que temos habilitado, podemos ter mais de um servidor habilitado. Ou seja, uma máquina que disponibiliza os pacotes binários para o pacman pode conter mais do que um repositório, por exemplo *extra* e *core*. Veja exemplo abaixo:

Configuração simples do pacman.conf:

```
[core]
Server=http://archlinux.c3sl.ufpr.br/core/os/i686
```

Configuração dinâmica:

```
[core]
Include=/etc/pacman.d/mirrorlist
```

E um exemplo do arquivo *mirrorlist*, gerado pela ferramenta citada logo abaixo:

```
##
## Arch Linux repository mirrorlist
## Sorted by mirror score from mirror status page
```

```
## Generated on 2010-11-23
## Score: 5.4, Brazil
Server = http://archlinux.c3sl.ufpr.br/$repo/os/$arch
## Score: 6.8, Brazil
Server = http://www.las.ic.unicamp.br/pub/archlinux/$repo/os/$arch
## Score: 7.8, Brazil
Server = ftp://archlinux.c3sl.ufpr.br/archlinux/$repo/os/$arch
## Score: 9.1, Brazil
Server = http://pet.inf.ufsc.br/mirrors/archlinux/$repo/os/$arch
## Score: 10.6, Brazil
Server = ftp://ftp.las.ic.unicamp.br/pub/archlinux/$repo/os/$arch
```

O que ganho com isto ? Ganho em atualização, pois um pacote de por ventura não estiver ainda atualizado em um servidor A, pode estar atualizado no servidor B. Ganho em disponibilidade, pois um pacote pode ainda não existir em um servidor A mas existir no servidor B. Ganho em performance, pois temos como ver quais são os servidores mais rápidos, em questão de resposta e download dos arquivos e deixá-los como primeiros na lista. O padrão da distribuição é já conter um arquivo `/etc/pacman.d/mirrorlist` contendo muitos servidores espelho, e o `pacman.conf` já apontando para este arquivo. Porém, o método mais interessante de ajustar a nossa lista de servidores é utilizando a ferramenta online, no site do ArchLinux.org, chamada "mirrorlist" (<http://www.archlinux.org/mirrorlist/>).

Mirrorlist

Acessando a ferramenta, é fácil entender como funciona, basta selecionar o país (no nosso caso Brasil), o protocolo (pode deixar os dois, http e ftp, marcados), a versão do protocolo ip que deseja e marque "Use mirror status" para que seja gerada uma lista ordenada pelo tempo de resposta dos servidores.

Gerada a lista, copie-a e grave sobre o arquivo `/etc/pacman.d/mirrorlist` e assim você terá uma lista organizada da melhor forma possível.

Além de que, podemos também acompanhar os resultados de estatísticas geradas periodicamente sobre a lista de mirrors oficial em <http://www.archlinux.org/mirrors/status/>.

Estabilidade

O que geralmente acontece, principalmente para se manter uma instalação bem estável do Arch é habilitar somente os repositórios core e extra, e as vezes o community. Assim não ficamos com pacotes mais instáveis ou de teste, gerando erros inesperados, a não ser que você esteja querendo ajudar no teste dos pacotes, para reportar os bugs quando necessário, o que é muito bom para os desenvolvedores se seus pacotes forem testados na maioria de computadores possível.

Ordem dos Repositórios

Outro detalhe é que a ordem em que os repositórios aparecem no `pacman.conf` é muito relevante, pois, o `pacman` instalada sempre o pacote do primeiro repositório que o contenha, ou seja, digamos que temos o `gnome 2.28` no core e o `2.30` no testing, se você tem os dois repositórios habilitados no seu `pacman.conf`, mas o core na frente, e solicitar a instalação do `gnome` ele vai instalar o `2.28`, porém se você inverter a ordem dos repositórios, ele irá instalar o `2.30`. Sendo assim, você até pode habilitar o testing como o último repositório da sua lista, e quando necessitar instalar uma versão de pacote que esteja lá, pode instalá-lo especificando o repositório no comando `: pacman -S testing/gnome`.

Alguns comandos úteis do Pacman

- Procurar por pacotes: `pacman -Ss <pedaço ou nome completo do pacote>`
- Instalar pacote: `pacman -S <nome_do_pacote>`
- Remover pacote: `pacman -R <nome_do_pacote>`
- Remover pacote e suas dependências: `pacman -Rs <nome_do_pacote>`
- Remover pacote e suas configurações: `pacman -Rn <nome_do_pacote>`
- Atualizar o sistema: `pacman -Syu`
- Lista de pacotes instalados no sistema: `pacman -Q`
- Lista de pacotes órfãos: `pacman -Qtd`

Estatísticas

A pouco tempo atrás foi liberada uma ferramenta para a comunidade ArchLinux que ajuda muito o trabalho dos líderes e TU do projeto: **pkgstats**

Bom, seu nome diz tudo: estatísticas de pacotes. Simples assim! Esta ferramenta envia todos os dados referentes aos seus pacotes e repositórios utilizados, para o servidor do Arch, onde estes, são consolidados e atualizados de tempos em tempos.

Veja você mesmo quais pacotes de repos são mais ou menos utilizados e outras informações que você possa concluir como um bom investigador: <https://www.archlinux.de/?page=PackageStatistics>

E neste link (<https://www.archlinux.de/?page=FunStatistics>) tem algumas estatísticas em modo resumido e mais claras !

Para ajudar com estes dados basta você instalar o programinha *pkgstats*, que encontra-se no repositório *extra*. Depois só executar !

2.4. Como criar um pacote para o Arch Linux

O pacman trabalha com arquivos binários para instalar os programas, mas para gerar um binário precisamos primeiro criar um pacote. Qualquer usuário de Arch pode montar um pacote destes. Montar o pacote nada mais é do que: informar ao sistema como montar este pacote, especificando todos os passos e arquivos necessários para que um programa possa executar corretamente, um PKGBUILD.

Tah, mas como fazemos isto ?

O mestre aqui é o **PKGBUILD**! Este é o cara! É nele que inserimos todas as informações para a geração. Quando você executa o comando *makepkg*, ele irá procurar por um arquivo PKGBUILD no diretório atual, e compilará o código fonte do programa de acordo com as instruções contidas neste arquivo. O pacote binário gerado terá o seu nome no formato: *nomedopacote-versão-release-*

arquitetura.pkg.tar.gz, o qual poderemos instalar depois com o comando `pacman -U nomedopacote-versão-release-arquitetura.pkg.tar.gz`.

Uma observação interessante: É aqui que se aplica a afirmação: O ArchLinux utiliza um sistema de geração *ports-like*. Então vamos utilizar a árvore de trabalho do ABS (Arch Build System), que fica em */var/abs/local*.

Então vamos lá! Utilizaremos aqui, como exemplo somente, o pacote "nginx"¹.

Primeiro necessitamos instalar todas as ferramentas necessárias para o desenvolvimento do nosso pacote, executem o comando abaixo:

```
pacman -S base-devel
```

Além de, é claro, tudo que for necessário para poder compilar e testar nosso programa.

Pronto! Estamos com as ferramentas devidamente instaladas! :-) Agora você deve testar se a instalação do seu programa está funcionando.

Por exemplo, execute:

```
./configure
make
make install
```

Funcionou ? Beleza! Então criem um diretório com o nome do seu pacote em */var/abs/local*, por exemplo

```
mkdir /var/abs/local/nginx
cd /var/abs/local/nginx
```

Então agora vamos criar o nosso arquivo PKGBUILD. Ele é um simples arquivo de texto, porém formatado de maneira que a ferramenta o entenda e gere nosso pacote correto. Para começar um novo PKGBUILD, você pode utilizar um protótipo que tem disponível no sistema já: */usr/share/pacman/PKGBUILD.proto*

Veja-o abaixo:

```
# This is an example PKGBUILD file. Use this as a start to creating your own,
# and remove these comments. For more information, see 'man PKGBUILD'.
# NOTE: Please fill out the license field for your package! If it is unknown,
# then please put 'unknown'.

# Maintainer: Your Name <youremail@domain.com>
pkgname=NAME
pkgver=VERSION
pkgrel=1
pkgdesc=" "
arch=( )
url=" "
license=('GPL')
groups=( )
depends=( )
makedepends=( )
optdepends=( )
```

¹Nginx não é um programa criado pelo autor deste livro, foi somente utilizado como exemplo

```
provides=()
conflicts=()
replaces=()
backup=()
options=()
install=
changelog=
source=($pkgname-$pkgver.tar.gz)
noextract=()
md5sums=() #generate with 'makepkg -g'

build() {
  cd "$srcdir/$pkgname-$pkgver"

  ./configure --prefix=/usr
  make
}

package() {
  cd "$srcdir/$pkgname-$pkgver"

  make DESTDIR="$pkgdir/" install
}
```

Vamos ver os detalhes deste arquivo PKGBUILD:

pkgname

O nome do pacote. Alfanumérico somente. Letras devem ser em minúsculo.

pkgver

A versão do pacote. Deve ser a mesma versão do software em questão. Pode conter letras e números. Hífen não é permitido, se caso necessário, deve ser trocado por underline.

pkgrel

O número do release deste pacote exclusivo para o ArchLinux

pkgdesc

Descrição do pacote. Pode conter até 80 caracteres somente. Por convenção não deve conter o nome do pacote novamente.

arch

Uma lista das arquiteturas suportadas pelo programa. Ex: arch=(*i686 x86_64*)

url

A url do site oficial do programa

license

A licença pela qual o software é distribuído

groups

A qual grupos de pacotes este pacote pertence. Por exemplo o *kdebase*

depends

Lista de pacotes de devem ser instalados antes de instalar este pacote. Não é necessário listar aqui pacotes que já sejam dependências das atuais dependências. Ex: *gtk2* depende de *glib2* e *glibc*, porém só liste aqui *glib2*, pois *glibc* já é dependência de *glib2*.

makedepends

Lista de pacotes que são necessário para a geração do pacote, mas não para a utilização do mesmo. Itens de base-level não necessitam ser listados aqui

optdepends

Lista de pacotes que não são obrigatórias para o funcionamento do programa, mas podem adicionar funcionalidades caso instaladas

provides

Uma lista de pacotes dos quais este pacote proverá funcionalidades. Ex: Se você criou uma versão do qt, chamdo *qt-foobar* liste aqui o pacote: *qt*

conflicts

Lista de pacotes que podem causar problemas caso estejam instalados simultaneamente

replaces

Lista de pacotes obsoletos, aos quais este pacote será utilizado no lugar.Ex: Lista *ethereal* para o pacote *wireshark*

backup

Lista de arquivos que deve ser salvos como *<arquivo>.pacsave* para que não sejam perdidos na remoção. Comumente utilizado para pacotes que alteram configurações no */etc*

options

Uma lista de opções que permite alterar alguns itens do comportamento padrão do comando *makepkg*:

- **strip**: Retira símbolos dos binários
- **docs**: Salva o diretório */docs*
- **libtool**: Deixa as libtool (arquivos *.la*) no pacote
- **emptydirs**: Deixa os diretórios vazios no pacote
- **zipman**: Comprime as páginas *man* e *info* com *gzip*
- **ccache**: Permite utilização de *ccache* durante a geração
- **distcc**: Permite utilização de *distcc* durante a geração
- **makeflags**: Permite o uso de *makeflags* específicas do usuário
- **force**: Força ao pacote ser atualizado pelo sistema de atualização do pacman.

install

O nome do script de instalação (*.install*) que deve ser executado. Este deve ser incluído no pacote. O pacman tem a habilidade de executar scripts específicos dependendo da operação executada, estes procedimentos são definidos como funções (Um protótipo deste arquivo pode ser encontrado em */usr/share/pacman/proto.install*), vejam:

- **pre_install**: Executa antes dos arquivos serem extraídos. Recebe o número da nova versão.

- **post_install**: Executa depois dos arquivos serem extraídos. Recebe o número da nova versão.
- **pre_upgrade**: Executa antes dos arquivos serem extraídos. Recebe o número da nova versão e o número da antiga versão.
- **post_upgrade**: Executa depois dos arquivos serem extraídos. Recebe o número da nova versão e o número da antiga versão.
- **pre_remove**: Executa antes dos arquivos serem removidos. Recebe o número da antiga versão.
- **post_remove**: Executa depois dos arquivos serem removidos. Recebe o número da antiga versão.

source

Lista de arquivos que serão necessários para a geração. Deve conter a localização dos arquivos, e na maioria dos casos é um endereço HTTP:// ou FTP://. As variáveis *pkgname* e *pkgver* podem ser utilizadas aqui.

noextract

Lista de arquivos, dentro dos pacotes em *source*, que não devem ser extraídos

md5sums

Lista das somas MD5 respectivos aos arquivos informados no *source*, que servirão para comparação após o download

sha256sums, sha384sums, sha512sums

Lista de somas SHA-2 com 256, 384 e 512 bytes. Alternativas ao MD5sum.

Ok, agora que já conhecemos as opções do nosso pacote, que chamamos de meta-configurações, vamos passar para o próximo passo. Temos que construir a função "build()" dentro do PKGBUILD.

Função build

Esta função não faz nada mais do que executar todos os passos necessários para a compilação/geração do seu programa. Sendo que todo o resultado desta operação será empacotado automaticamente depois. Nela utilizamos comandos comuns do shell bash, temos disponíveis as variáveis *\$srcdir* que contém o caminho onde está o PKGBUILD e todas as outras variáveis que foram definidas acima, nas meta-configurações do PKGBUILD.

O primeiro passo é entrar no diretório criado pela descompressão dos fontes "*\$srcdir/\$pkgname-\$pkgver*". Assim saberemos onde estamos, e devemos trabalhar com caminhos relativos. Agora você vai utilizar os mesmos comandos utilizados para compilar o programa manualmente. Na maioria dos casos você vai utilizar algo como: `./configure; make`, apesar de existirem centenas de variações para compilar os programas, como por exemplo `ant build`, ou outro comando qualquer para compilar os fontes do seu programa. A função `build` automatiza tudo o que você faz manualmente, em um ambiente `fakeroot`. Se o software que você está compilando utiliza-se do script `configure` é uma boa prática utilizar `--prefix=/usr` neste comando, pois todos os pacotes gerenciados pelo `pacman` devem utilizar o diretório `/usr`. Deixe o diretório `/usr/local` para os pacotes gerados e instalados manualmente.

Função package

A parte final do empacotamento é colocar os arquivos compilados em um diretório onde o `makepkg` possa pegá-los para empacotar: o diretório `pkg`.

Este diretório é simplesmente um ambiente *fakeroor*, ele deve basicamente imitar o seu filesystem root para o procedimento de instalação do programa. Quaisquer arquivos que devam ser instalados/copiados em um diretório do seu filesystem devem estar no diretório `pkg` na mesma estrutura de diretórios

O comando `makepkg --repackage` roda somente a função *package*, sem executar novamente a compilação do programa. Isto facilita bastante, além de não gastar tempo em vão, quando queremos alterar somente algo que não interfira na compilação do mesmo, como por exemplo as dependências do pacote.



Dica

O comando `makepkg`, e também a função `build()`, devem funcionar de maneira não-interativa. Utilitários ou scripts interativos executados na função `build()` podem quebrar o `makepkg`, principalmente se ele for chamado com a opção `build-logging` habilitada (`-l`).

Agora, se for necessário executar alguns passos após o processo de instalação, ou mostrar mensagens ao usuário, utilizamos o arquivo `.install`. Lembre-se que definimos qual o arquivo `.install` na configuração "install" definida acima.



Importante

Para gerar um pacote que seja bem aceito na distribuição, temos que seguir algumas regrinhas:

- Os pacotes **NUNCA** devem ser instalados em `/usr/local`
- Não crie novas variáveis nos scripts do `PKGBUILD`, pois podem gerar erros e conflitos durante o processo. Caso seja **extremamente** necessário, crie-a prefixada com um underline
- A variável `packager` do meta-file pode ser customizada alterando-a no arquivo ou então com uma variável de ambiente chamada `PACKAGER`.
-
- **TODAS** as mensagens importantes devem ser mostradas utilizando o arquivo `.install`.
- Quaisquer dependências que não sejam obrigatórias para o funcionamento do programa ou tenham funções gerais, devem ser incluídas no `optdepends`
- Quando criar a descrição do pacote, não repita o nome do programa. Ex, ao invés disto: "Nginx é um servidor web", use isto: "Servidor web"
- Tente deixar as linhas do `PKGBUILD` com menos de 100 caracteres.
- Quando possível, remova as linhas vazias ou com variáveis sem valor.
- Preserve a ordem dos campos do `PKGBUILD`
- Veja as regras completas em: https://wiki.archlinux.org/index.php/Arch_Packaging_Standards

Ok, agora vamos ter que testar o nosso pacote né ! Para isto basta rodar o comando `makepkg` no diretório onde se encontra o PKGBUILD criado, no nosso caso em `/var/abs/local/nginx` (seguindo o nosso exemplo!). Se estiver tudo correto, será criado um pacote, com seu nome no formato `$pkgname-$pkgver.pkg.tar.xz`, porém, se der algum erro, será mostrado uma mensagem de erro. Daí você terá que avaliar o que aconteceu, se faltou algum passo para a geração, se informou algum parâmetro errado, ou se faltou alguma biblioteca, ou se criou alguma inconsistência no PKGBUILD mesmo. Avalie, arrume, e volte a executar o `makepkg` até que dê tudo certo, ok !

O pacote gerado pode ser instalado com o comando `pacman -U`, por exemplo: `pacman -U nginx-0.9.pkg.tar.xz`.

Agora, para completar o serviço temos duas ferramentas que podem nos auxiliar em verificações dos nossos pacotes:

- **ldd** : Mostra quais as dependências das bibliotecas compartilhadas de executáveis.
- **namcap** : É uma ferramenta mais completa, que verifica as dependências, os binários e o código fonte dos pacotes, procurando por erros que possam danificar ou agredir o seu sistema, além de alguns erros que possam existir nos arquivos PKGBUILD. Veja mais sobre ele aqui: <https://wiki.archlinux.org/index.php/Namcap>

Depois de gerar este pacote, e de ele funcionar, o que devemos fazer com ele ? Guardar em um CD ? Colocar no pendrive ? **Não** mesmo! Vamos compartilhá-lo com a comunidade! Não foi para isto que você criou um pacote do seu programa !!!

Mas como ? Vamos enviá-lo ao AUR (*ArchLinux User Repository*).

2.5. Como enviar pacotes para o AUR

Depois de ter criado seu pacote, testado, e tudo mais, então poderás enviá-lo ao repositório de pacotes de usuários do Arch Linux, o AUR.

Para fazer isto crie um pacote com tudo necessário, do diretório no formato do ABS, por exemplo: `/var/abs/meupacote`, contendo o arquivo `PKGBUILD`, os arquivos `.install`, quaisquer patches, etc... Nada binário deve ser enviado, somente arquivos fonte.

Para isto, execute o comando `makepkg --source`, assim o pacote correto para o envio ao AUR será criado.

- Antes de enviar seu pacote verifique alguns itens:
 - Verifique os repositórios que contém pacotes binários (*core, extra, community*, etc). Se encontrar o pacote lá **NÃO** envie-o ao AUR. Caso tenha algum problema, abra um bug no bugtrack do Arch.
 - Verifique se o pacote está presente e sendo mantido por alguém no AUR mesmo. Caso positivo, envie sugestões pelo sistema de comentário. E se estiver orfão, é possível adotá-la.
 - Verifique sempre, e de novo, e com atenção os padrões de empacotamento do Arch Linux (https://wiki.archlinux.org/index.php/Arch_Packaging_Standards), isto é essencial para o sucesso e bom funcionamento da distribuição.

- Pacotes com binários ou muito ruins serão excluídas sem aviso prévio.
- Se você tem dúvidas sobre seu pacote, utilize a lista de discussão ou o fórum para pedir ajuda e para que alguém faça uma verificação antes de enviar definitivamente.
- Saiba se este pacote é realmente interessante! É muito especializado ? Muitas pessoas gostarão de ver este pacote disponível ? Quanto mais melhor...
- Obtenha experiência antes de enviar o pacote fazendo várias compilações para conhecer bem os métodos e depois sim envie seu pacote.
- Se você enviar um o arquivo "meupacote.tar.gz" contendo um arquivo "meupacote", você receberá um erro "Could not change to directory /home/aur/unsupported/meupacote/meupacote". Para resolver este problema, renomeie o arquivo para meupacote.rc, por exemplo, e depois na instalação você pode renomeá-lo para "meupacote" novamente.

Então vamos lá, verificado tudo, faça (ou crie) o login no site do AUR (<http://aur.archlinux.org/>) e clique no menu *enviar*, como ilustrado na imagem abaixo.



Depois selecione a categoria, informe o caminho do arquivo compactado e clique em "Enviar".

Feito o envio, parabéns !



Ajudinha

Caso você não queira (ou não possa) utilizar a interface web para enviar o pacote, utilize um dos dois programas que tem para auxiliar nesta tarefa: auruploader ou burp (procure no AUR)

Capítulo 3. ABS/AUR

3.1. Arch Build System (ABS)

(ou somente ABS), assim é chamado o sistema de geração de pacotes que o ArchLinux utiliza. O ABS, é um sistema ports-like, ou seja, é parecido com o sistema "ports" de geração de pacotes, utilizado nos sistemas BSD. Com ele é possível que o código fonte do referido pacote seja baixado, desempacotado, compilado e instalado automaticamente. Porém, ao invés de utilizar apenas um diretório para efetuar este processo, como no port, o ABS utiliza-se de um arquivo chamado PKGBUILD, que contém todas as informações para poder baixar/compilar/installar o tal pacote.

Quem utiliza o Arch não tem a obrigação de conhecer nem utilizar o ABS. Pois ele é uma ferramenta de auxílio para os que queiram ajustar ou manipular ainda mais seu sistema, customizando, baixando e compilando vários programas conforme suas necessidades.

3.2. ArchLinux User Repository (AUR)

(ou somente AUR) é um repositório de PKGBUILDS mantido pela comunidade. Ou seja, são informações de compilação para os mais diversificados programas que não estão no repositório oficial do Arch. Por exemplo, temos o pacote "firefox-nightly", que referencia a geração diária do Firefox, e obviamente, não estará no repositório oficial, justamente por ser um build de desenvolvimento. Mas alguém dedicado, foi lá e criou o arquivo PKGBUILD, direcionado a este software, e o adicionou no AUR, disponibilizando-o para que nós, pudéssemos baixar o arquivo e compilá-lo para nosso sistema facilmente.

Tá, mas aí você pensa assim: - Poxa, é legal isto, mas dá um trabalhão pegar um pacote deste AUR, não ! Até posso dizer que não é algo trivial, só que como sempre na informática, alguém com certeza já passou por este mesmo problema.

YAOURT e *aurget* são duas ferramentas que interagem com o AUR, funcionalidade que o Pacman não faz, pois este é destinado somente aos pacotes binários, por isto a necessidade de se utilizar uma ferramenta diferente.

A diferença básica entre *aurget* e *YAOURT* é que um é um script BASH e o outro um programa compilado. Ambos fazem bem o seu papel, porém o *yaourt* está mais bem difundido e é a mais completa delas, na minha opinião, fazendo inclusive validação de dependências e conflitos.

Se você tiver o programa "aurvote" instalado, o yaourt automaticamente verifica se você já votou no pacote instalado, e se não votou o questiona para votar (Veja mais adiante sobre Votação de Pacotes).

Vale lembrar também que, além destes dois programas, tem outras opções para se trabalhar com o AUR, basta dar uma procurada na lista de programas do AUR mesmo, via web.

3.2.1. Yaourt

Vamos instalar o yaourt para trabalhar com os pacotes do AUR ?

Adicione o seguinte repositório no seu arquivo "pacman.conf", atentando para a sua arquitetura

```
[archlinuxfr]
#Server = http://repo.archlinux.fr/i686
#ou
#Server = http://repo.archlinux.fr/x86_64
```

Agora basta atualizar e instalar facilmente com o pacman:

```
$ sudo pacman -Sy yaourt
```

Então estamos prontos para utilizar o yaourt para nossos exemplos:

Observação Importante: Sempre utilize o yaourt como usuário normal, nunca como root, evitando assim, que qualquer script malicioso tenha acesso total ao seu sistema. E caso você tenha dúvida sobre o que está ocorrendo ou vai ocorrer, abra o arquivo PKGBUILD do pacote para averiguar, para quem conhece Bash é simples descobrir o que a instalação do pacote faz. Certo !

Então tá, vamos lá:

```
yaourt <pacote>
    Procurar por um pacote
```

```
yaourt -S <pacote>
    Instalar pacote
```

```
yaourt -Syu --devel --aur
    Atualização do sistema de dos pacotes instalados pelo aur
```

Com isto, você já poderá se virar e utilizar os pacotes criados pelos usuários, e assim usufruir muito do seu Arch.



Uma nota pessoal

Em praticamente 3 anos de utilização do Arch, se necessitei instalar 4 ou 5 pacotes que **não** estivessem disponíveis no AUR, foi muito. O resto TUDO eu encontro por lá! Só para vocês terem uma ideia do poder do AUR e seus usuários.

Veja mais em: <http://archlinux.fr/yaourt-en>

3.2.2. Enviar Pacotes para o AUR

Mas eu posso enviar pacotes para o AUR ? **SIM** !! Obviamente que SIM ! Pois o AUR é um repositório de PKGBUILDS feito pela comunidade de usuários! Porém, não ache que é simplesmente enviar qualquer coisa, os pacotes para serem bem aceitos tanto pelos usuários do Arch quando pelos TU, para possivelmente chegarem ao repositório [community] devem ser pacotes confiáveis, sem truques, sem vírus e que sigam os padrões de construção! Assim você será um ótimo contribuidor!

Enviar o seu PKGBUILD para o AUR, também conhecido como *[unsupported]*, é simples, veja a instrução completa no item Seção 2.5, “Como enviar pacotes para o AUR”.

3.2.3. Votação de Pacotes

Os *pacotes* que estão no AUR não são considerados pacotes ainda, eles são apenas PKGBUILDS. Estes pacotes são criados pelos usuários que não são TUs (Trusted Users Veja no próximo item

quem são os TUs) e disponibilizados para que todos que utilizem o ArchLinux possam instalá-los e utilizá-los com facilidade. Porém cada pacote disponível no AUR pode um dia vir a ser adotado por um TU e adicionado aos repositórios oficiais da distro. E para que isto seja possível, eles levam em consideração o número de votos!

A votação é simples, é uma eleição de pacotes: os que foram mais votados é porque mais usuários que utilizaram o pacote gostaram dele, e deram-no seu voto de confiança, seu voto de OK! Votando você diz que gostaria que este pacote esteja disponível no repositório oficial, e assim, instalável via `pacman`. Mas existe uma regra básica: somente pacotes que recebem 25 votos ou mais, poderão ser adotados por um TU e seus binários incluídos no repositório `community`.

Mas como eu faço para votar em um pacote ?

O modo principal para votação de pacotes é pela web. Você vai até o endereço `http://aur.archlinux.org/packages.php`, efetua o login, busca pelo pacote desejado, clica sobre o nome do escolhido, e, logo abaixo das informações do pacote, você verá um botão "Votar", clicando neste botão você terá votado no seu pacote favorito!

Porém, outro modo de votar é por um script chamado "*aurvote*". Ele tem seu PKGBUILD disponível no AUR também. Portanto baixe-o e instale normalmente. Então para votar com ele basta o seguinte comando:

```
$ aurvote --vote <pacote>
```

Ok ! Pacote votado, e você já é um contribuidor do ArchLinux !

3.3. Trusted Users

TRUSTED!

A denominação já diz tudo: Usuários de Confiança, ou, Usuários Confiáveis! São pessoas da comunidade, em torno do Arch, que ganharam a confiança de todo o time. Você não compra um lugar de TU, ou, faz uma prova e vira TU, não é uma certificação, mas sim, um reconhecimento pelo trabalho bem feito.

Mas porquê?

Justamente por que para se tornar um TU você precisa ser indicado pelas pessoas da comunidade para tal função, e então o time dos atuais TUs irá votar, de forma democrática, se quer ou não você como TU!

Mas o que faz um TU?

Os TUs são responsáveis pela boa organização e funcionamento do AUR. Eles que averiguam se um pacote enviado para o AUR, contém códigos maliciosos, e gerenciam os pacotes para que, através das regras definidas, entrem ou não no repositório [*community*], o qual é responsabilidade dos mesmos TUs, fora que, eles também tem direito de voto em questões administrativas do projeto.

3.4. Compilando através do ABS

O ABS é o nosso amigo! Ele facilita e muito a criação e instalação de pacotes, pelos códigos fonte, no nosso sistema. Não utilizá-lo é perder tempo com tarefas padrão.

Vamos ver ?

- O ABS é utilizado para
 - Compilar ou recompilar um pacote por quaisquer razões
 - Criar e instalar novos pacotes através dos fontes se estes não estiverem disponíveis no AUR.
 - Customizar pacotes conforme suas necessidades
 - Recompilar todo o seu sistema, utilizando suas flags de compilação, "a la FreeBSD"
 - Gerar e instalar seu próprio kernel
 - Gerar e instalar módulos no seu kernel customizado (ou não)
 - Facilmente compilar e instalar um pacote do AUR mais novo, beta ou em desenvolvimento simplesmente editando o número da versão no PKGBUILD.

Vamos disponibilizar o ambiente do ABS para trabalhar:

```
# pacman -S abs base-devel
```

Vai ser criada uma árvore de diretórios em `/var/abs`, e ela tem a mesma estrutura do banco de dados de pacotes, contendo somente os arquivo PKGBUILD e não os arquivos fontes. Estes serão baixados na hora da compilação, da URL definida no PKGBUILD.

Algo assim:

- `/var`
 - `/abs`
 - `/<branch>`
 - `/<categoria>`
 - `/<nome_do_pacote> /PKGBUILD`

Para sincronizar um pacote local com o remoto, utilize o comando:

```
# abs <repositorio>/<pacote>
```



Download da árvore

O primeiro download feito pelo ABS é o maior, sendo que ele baixa tudo. Porém ainda assim não é algo inconcebível, gira, atualmente, em torno de 16 a 20 MB de download, todo o conteúdo é somente texto e ainda assim compactado na fonte.

Em `/etc/makepkg.conf` você poderá editar as variáveis de ambiente e flags de compilação desejadas e conforme seu sistema, caso queira. As otimizações para arquiteturas `i686` e `x86_64` já estão ok para processadores monoprocesados (apenas 1 core), as configurações padrão irão funcionar corretamente em processadores SMP (multiprocessados), porém irão compilar com somente 1 core.

O indicado para efetuar compilações e modificações é que você crie um diretório só seu para fazer isto, pois quando for sincronizada a árvore padrão (`/var/abs`) todas as alterações serão perdidas. Então faça um dos dois itens abaixo:

- Criar o diretório em `$HOME/abs`
- Criar o diretório em `/var/abs/$USER` e dar permissões totais para `$USER`.

Agora, o `PKGBUILD` do pacote desejado de `/var/abs` para seu diretório:

```
# cp -R /var/abs/<branch>/<categoria>/<pacote>/PKGBUILD /path/to/my/abs/<pacote>
```

depois edite o `PKGBUILD` para alterar algo desejado, compile e instale:

```
# cd /path/to/my/abs/<pacote>
# vim PKGBUILD
# makepkg -s
# pacm -U meu_pacote_gerado_i686_.pkg.tar.xz
```

Pronto!

Mas fica ai a dúvida de alguns:

Mas eu não posso fazer o download direto do `PKGBUILD`, com o `wget` por exemplo, e somente alterar o que quero e efetuar a mesma compilação ?

Sim, pode, porém tendo a árvore do ABS localmente temos algumas vantagens:

- Podemos efetuar compilações desconectados da internet (obviamente se tivermos os arquivos fonte)
- Podemos mais facilmente alterar e recompilar qualquer pacote, mesmo dos repositórios core e extra por exemplo.
- Necessitaremos dela para efetuar uma recompilação geral do sistema.



Makepkg.conf

Você pode encontrar no Wiki do Arch Linux maiores informações sobre as diretivas de compilação, e além disto, pode acrescentar seu conhecimento para melhorar ainda mais a compilação e/ou utilização de um pacote.



Tudo?

Se você for mais atrevido, e gosta de situações perigosas e desafiadoras, você pode utilizar o script `pacbuilder` [<https://wiki.archlinux.org/index.php/Pacbuilder>] tanto para recompilar algum programa específico, ou, os programas de um repositório específico ou até TODO o sistema!

Capítulo 4. Mais informações e contribuições

4.1. Projetos Relacionados

O projeto ArchLinux tem suas regras e condutas bem definidas, basta ler o artigo "O Jeito Arch"¹. E a que mais se ressalta é a simplicidade. Só que isto não agrada a todos, pois nem todos veem simplicidade do modo que o Arch vê!

Geralmente os usuários de computador, seja de qual sistema operacional for, veem simplicidade como "não ter que utilizar o terminal de comando", "ter programas que auxiliam na configuração de tudo", "tarefas automatizadas e automáticas", "ambientes completamente auto-configuráveis", e por aí vai. Mas no arch não é bem assim, simplicidade aqui é transparência, é boa documentação, é fácil acesso, é não esconder o que tem por trás das operações.

Mas como o Arch também é um projeto open-source, como a maioria das distros, tem alguns usuários programadores que criam algumas facilidades para os usuários mais novatos, que também tem direito de aprender o *Jeito Arch de Ser*¹, mas mais devagar, não tão bruscamente ! hehe :-)

E estes projetos dão uma contribuição enorme ao ArchLinux, uma vez que aprimoram a sua utilização para várias pessoas e divulgam ainda mais a distro.

Vamos conhecer agora, alguns destes projetos que trazem diversidade e conhecimento para nós, e depois, mostrarei onde encontrar mais projetos relacionados:

Archbang

O ArchBang é um projeto de LiveCD, que roda um ambiente ArchLinux completo, leve, com OpenBox window manager, servindo tanto para rodar e instalar em seu desktop como num sistema portátil, como um pendrive.

Site: <http://archbang.org/>

Godane ArchIso-live

Este projeto é do mesmo gênero do ArchBang, porém utiliza o Gnome como ambiente de desktop manager. Tem um bom movimento, gerando praticamente 1 release do livecd por mês.

Site: <http://godane.wordpress.com/>

Chakra

Com a mesma intenção de ambos já citados, o projeto Chakra provê um ambiente completo para desktop, todo encima do ArchLinux, porém utilizando o belíssimo ambiente KDE4. Porém este projeto provê também algumas ferramentas a mais como o Tribe, um instalador gráfico bem bonito e facilitado, o CInstall, que é o gerenciador de pacotes além do LiveScripts2 que é um grupo de ferramentas para (re)criação de ambientes live como CD e Pendrives.

¹O Jeito Arch - Link [https://wiki.archlinux.org/index.php/The_Arch_Way_%28Portugu%C3%AAs%29]

Site: <http://chakra-project.org>

ArchAudio

ArchAudio é um projeto relacionado ao ArchLinux voltado à gravação, mixagem, masterização, design de som e programação DSP. Ele disponibiliza os repositórios com todos os programas e ferramentas relacionadas e necessárias para estas atividades.

Site: <http://archaudio.org/>

ArchServer

Um Arch Linux com atualizações mais controladas e diferentes mirrors de atualização, para ambientes de servidor.

Site: <http://www.archserver.org/>

CTKArch

É uma distribuição minimalista, onde o seu criador selecionou alguns pacotes simples e úteis para um uso rápido. É um liveCD que pode ser instalado ou não em seu disco rígido. Ele pode ser gravador em um cartão de memória/pendrive/cd. Utiliza os mirrors oficiais do Arch Linux.

Site: <http://ctkarch.org/>

E para encontrar mais projetos relacionados ao ArchLinux você pode acessar esta página: https://wiki.archlinux.org/index.php/Related_Projects

Isto tudo, sem contar ainda os projetos que são enviados ao AUR. Este sim é o verdadeiro berço de inovações e criatividade da comunidade Arch. Nele encontramos praticamente tudo que precisamos para nosso ambiente. Pacotes são inseridos e atualizados todos os dias. Então, nunca deixe de verificar no AUR se o software que você quer existe lá, provavelmente sim! E senão tiver ? Empacote o programa e envie-o para lá ! Contribua você também!

Rolling Release

Rolling Release é uma plataforma de compartilhamento de informações que interessam aos archers. É uma eZine muito interessante, porém em Inglês. O endereço do site é: <http://rollingrelease.com/>

Capítulo 5. Perguntas Frequentes

Esta parte reproduzirá (em tradução livre) algumas das perguntas respondidas no FAQ oficial do site do Arch, principalmente sobre algum ponto que possa não ter sido abordado ou ter ficado tão claro neste livro.

5.1. P: Porque eu iria querer utilizar o Arch ?

R: Se, depois de você ler sobre a filosofia do Arch, desejar fazer você mesmo, e quiser uma simples, elegante, altamente customizável e genérica distribuição GNU/Linux você é um novo usuário Arch.

5.2. P: Porque eu não quero utilizar o Arch ?

1. Se você não concordar com a Filosofia Arch
2. Você não tem tempo, habilidade ou vontade de fazer você mesmo uma distribuição GNU/Linux
3. Você quer suporte a uma arquitetura que seja mais antiga que x86_64 e i686
4. Você quer e tem uma firme posição sobre uma distro que forneça apenas softwares que sejam open-source tal como definido na GNU
5. Você quer um sistema operacional que seja auto-configurável, e trata por padrão um conjunto completo de softwares e ambiente de trabalho na mídia de instalação.
6. "you do not want a bleeding edge, rolling release GNU/Linux distribution."
7. Você está contente com o seu sistema operacional atual
8. Você está procurando um sistema operacional que seja direcionado a outros usuários

5.3. P: Em qual distribuição o Arch é baseado ?

R: O Arch foi desenvolvido independente, criado a partir do zero e não é baseado em nenhuma outra distribuição. Antes de criar o Arch, Judd admirava e utilizava o CRUX, uma bela e minimalista distro criada por Per Lidén. Originalmente inspirado em ideias do Crux, o Arch foi criado do zero, e o pacman foi codificado em linguagem C.

5.4. P: O Arch é direcionado a ser utilizado como servidor, desktop ou estação de trabalho ?

R: O Arch não é direcionado para qualquer tipo de uso específico. No entanto, é direcionado para um tipo de usuário. Arch é feito para que usuários mais experientes e que gostem do "faça-você-mesmo", explorando muito e moldando o sistema para suas necessidades específicas. Arch pode ser utilizado para praticamente qualquer finalidade, e claro, o <http://archlinux.org> roda sobre um Arch!

5.5. P: Quando vai sair a próxima versão do Arch?

R: As versões do ArchLinux são meramente uma "foto" dos pacotes contidos nos repositórios, combinadas com várias modificações e características do instalador. O modelo "rolling-

release" deixa seu Arch sempre atualizado com um simples comando. Por este motivo, versões não são tão importantes no Arch, pois no sistema rolling-release cria-se novas versões dos pacotes tão logo suas atualizações sejam liberadas. Se você quer a última versão do Arch, você não precisa reinstalar nada, basta rodar o comando "pacman -Syu" e seu sistema estará atualizado por completo, e idêntico ao de uma nova instalação. E por isto também, você não precisa ficar aguardando novas versões do sistema para ter novas features, pois assim que sair a versão do pacote, basta executar o comando "pacman -Syu" para tê-las disponíveis.

5.6. P: O Arch Linux é uma distribuição estável? Vão ter muitas quebras?

R: O Arch Linux é tão estável quanto você o faz! Você é que faz o sistema, você o cria encima de uma base simples e você controla os upgrades do seu sistema. O usuário é totalmente responsável pela estabilidade do ambiente, onde é quem decide quando vai atualizar o sistema, quais repositórios irá utilizar, quais pacotes irá instalar, etc.

5.7. P: Porque o pacman não tem uma interface visual oficial ?

R: Por favor leia o "O Jeito Arch" ¹. A resposta é basicamente que o time de desenvolvimento do Arch não irá desenvolver uma. Sinta-se a vontade para utilizar uma das interfaces criadas pelos usuários.

Capítulo 6. Listagem de comandos úteis de modo geral para o ArchLinux

- **pacman**
 - Atualizar sistema: `pacman -Syu`
 - Atualizar listas de programas: `pacman -Sy`
 - Procurar pacote: `pacman -Ss <nome>`
 - Instalar pacote: `pacman -S <nome>`
 - Remover pacote: `pacman -R <nome>`
 - Remover pacote com dependências não utilizadas: `pacman -Rn <nome>`
 - Remover pacote com dependências não utilizadas e configurações: `pacman -Rns <nome>`
 - Listar pacotes instalados: `pacman -Q`
 - Procurar pacote instalados: `pacman -Q <nome>`
 - Ver informação de um pacote instalado: `pacman -Qi <nome>`
 - Listar arquivos contidos no pacote: `pacman -Ql <nome>`
 - Ver a qual pacote um arquivo pertence: `pacman -Qo /path/to/file`
- **yaourt**
 - Atualizar sistema: `yaourt -Syu`
 - Atualizar listas de programas: `yaourt -Sy`
 - Procurar pacote: `yaourt -Ss <nome>` OU `yaourt <nome>`
 - Instalar pacote: `yaourt -S <nome>`
 - Remover pacote: `pacman -R <nome>`
 - Remover pacote com dependências não utilizadas: `pacman -Rn <nome>`
 - Remover pacote com dependências não utilizadas e configurações: `pacman -Rns <nome>`
- **aurvote**
 - votar em um pacote:
- **serviços**
 - iniciar serviço: `/etc/rc.d/<nome> start`

Listagem de comandos úteis de modo geral para o ArchLinux

- parar serviço: `/etc/rc.d/<nome> stop`
- **abs**
 - sincronizar árvore: `abs`
 - sincronizar pacote: `abs <repositorio>/<pacote>`
 - compilar um pacote com PKGBUILD: `makepkg`
- **rankmirror**
 - efetuar a verificação dos mirrors mais rápidos :
 - `mv /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.bkp`
 - `rankmirrors -n 10 /etc/pacman.d/mirrorlist.org > /etc/pacman.d/mirrorlist`
 - `pacman -Syy`

Capítulo 7. Instalação e configuração de programas

Nesta parte do livro você vai ver como instalar algumas partes do sistema, como o sistema gráfico base, ambientes gráficos e o sistema de som. Assim, você poderá partir para a utilização do sistema e seguir configurando-o conforme gosta.



Dica

Todos os comandos para instalação devem ser executados como root ou através do sudo, a não ser que seja solicitado outro usuário.

7.1. Como instalar ambiente gráfico base

O ambiente gráfico base é o software no qual os gerenciadores gráficos como Kde ou Gnome vão se apoiar para funcionar corretamente, ele é conhecido como X Window System ou simplesmente Xorg (<http://www.x.org>). É um software open-source e mantido pela X.Org Foundation. Então, antes de instalar qualquer gerenciador de desktop, precisamos instalar este.

Para instalar execute o seguinte comando no terminal, como root ou com sudo:

```
# pacman -S xorg
```

Feito isto, poderemos passar para a instalação do gerenciador de desktop desejado, ou para a instalação do seu driver de vídeo compatível com sua placa 3D.

7.1.1. Instalação do driver NVidia

Basicamente a instalação do driver da Nvidia não é difícil, salvo algumas placas com características diferentes que talvez não se adaptem a versão do driver corrente.

Você somente necessita prestar atenção na versão da sua placa para instalar o driver apropriado:

- Para placas mais atuais, com GPU depois das GF e FX instale:

```
# pacman -S nvidia nvidia-utils
```

- Para placas mais antigas, do modelo FX:

```
# pacman -S nvidia-173xx nvidia-173xx-utils
```

- E para as placas antes dos modelos GForce 4 instale:

```
# pacman -S nvidia-96xx nvidia-96xx-utils
```

Isto porque os drivers da Nvidia não carregam as compatibilidades de placas mais antigas nas versões mais novas!

O passo seguinte é a configuração do arquivo `xorg.conf`, que diz ao Xorg como deve funcionar, mas temos uma ferramenta de ajuda para isto, que vem no pacote da Nvidia: `nvidia-xconfig`



Antes de executar este comando (ou outro qualquer que altere algum arquivo) sempre faça um backup, pois poderá assim voltar qualquer alteração feita.

Então, execute o `nvidia-xconfig`, ele irá detectar o hardware existente na máquina e irá alterar o arquivo `/etc/X11/xorg.conf` conforme necessário para o bom funcionamento da placa no ambiente gráfico.

Caso você tenha múltiplos monitores conectados a placa, execute o `nvidia-xconfig` conforme abaixo:

```
# nvidia-xconfig --twinview
```

Fora isto, caso tenha ainda algum problema para executar o ambiente gráfico com sua Nvidia, dê uma olhada no wiki: <https://wiki.archlinux.org/index.php/Nvidia>

Na instalação do driver, será compilado e instalado o módulo do kernel referente, então aconselho você a reiniciar o computador antes de iniciar uma sessão gráfica.

7.1.2. Instalação do driver ATI Catalyst

O driver da ATI tem um repositório (não oficial) criado para manter maior compatibilidade entre o driver e as últimas versões do kernel, e até mesmo porque a instalação manual deste é bem chatinha. Detalhe importante: este driver funciona para as placas acima da versão R6xx somente, placas mais antigas que isto devem utilizar o driver *xf86-video-ati*.

Então vamos a instalação do driver:

Primeiro, adicione o repositório abaixo no seu `/etc/pacman.conf`:

Para ambientes 32bits:

```
[catalyst]
Server = http://catalyst.apocalypsus.net/repo/catalyst/i686
```

Para ambientes 64bits:

```
[catalyst]
Server = http://catalyst.apocalypsus.net/repo/catalyst/x86_64
```

Agora a instalação dos pacotes:

- Atualizar repositórios

```
# pacman -Syu
```

- Instalar o driver

```
# pacman -Rd libgl
# pacman -Sy catalyst catalyst-utils
```

- Instalar um xorg que tem alguns patches de correção para este driver

```
# pacman -Rd xorg-server
# pacman -S xorg-server-backclear
```

- Alterar o `/etc/X11/xorg.conf` com a configuração correta e automática

```
# aticonfig --initial
```

E, também, caso tenha algum problema maior e não conseguir subir o X dê uma olhadinha na página do wiki: https://wiki.archlinux.org/index.php/ATI_Catalyst

7.2. Como instalar um ambiente GNOME

A instalação do Gnome é muito simples, o que mais demora mesmo é ir ajeitando-o para deixá-lo como você quer, se não for utilizá-lo com a aparência e programas padrão.

Então vamos lá. Depois de ter instalado o ambiente gráfico base vamos instalar o que é necessário para o Gnome funcionar:

```
# pacman -S gnome
```

Este comando instalará o mínimo necessário para o Gnome funcionar. Porém queremos mais que o mínimo, não é? Então:

```
# pacman -S gnome-extra
```

Este pacote, *gnome-extra*, contém vários outros recursos do Gnome, que deixarão ele mais completo e funcional ainda.

Outro pacote interessante de se instalar é o *gnome-system-tools*, que contém ferramentas de gerenciamento do computador e do ambiente do Gnome

```
# pacman -S gnome-system-tools
```

Agora, vamos aprimorar nossa instalação. O Gamin é um sistema de monitoramento de arquivos e diretórios, e utilizado pelo Gnome. Sua instalação é simples e não necessita configuração alguma.

```
# pacman -S gamin
```

Obs: Verifique se na linha dos DAEMONS, no `/etc/rc.conf` se existe *dbus* ou *fam*, então retire-os e aproveite para desinstalar o `fam` `pacman -Rd fam`, pois no lugar dele, agora está sendo utilizado o Gamin.

Também, vamos editar o arquivo `/etc/hosts` e verificar se tem a seguinte linha configurada desta maneira, trocando *YOURHOSTNAME* pelo nome da sua maquina, definida na instalação:

```
127.0.0.1    localhost.localdomain    localhost    YOURHOSTNAME
```

Se não tiver esta linha, acrescente-a ou edite-a, deixando igual a esta.

Agora, vamos instalar o gerenciador de login GDM, ele é o responsável por autenticar os usuários para a utilização do ambiente:

```
# pacman -S gdm
```

Então, será necessário colocá-lo também na lista dos daemons que iniciam na inicialização do sistema, para que assim não seja necessário executar nada manualmente quando iniciar a máquina, então edite o arquivo `/etc/rc.conf`:

```
DAEMONS=(syslog-ng network crond gdm)
```

E, ainda neste arquivo, adicione (caso não exista) a interface *lo* ao lado da(s) sua(s) interface(s) configuradas. Ficando mais ou menos assim:

```
lo="lo 127.0.0.1"  
eth0="..."  
INTERFACES=(lo eth0)
```

Então, com isto temos o gerenciador de desktop Gnome rodando perfeitamente, e funcional para muitas atividades.

7.3. Como instalar um ambiente KDE 4

Para instalar o KDE, na versão 4.5 atualmente, você deve fazer o seguinte:

```
# pacman -S kde phonon-xine
```

ou

```
# pacman -S kde-meta phonon-xine
```

agora, para instalar a linguagem corretamente:

```
# pacman -S kde-l10n-pt_br
```



O KDE 4 é Modular, assim você pode instalar suas aplicações preferidas sem ter que instalar um monte de outras libs

Então, adicione o Dbus e o Hal na lista de daemons do */etc/rc.conf*, ficando mais ou menos assim:

```
DAEMONS=(syslog-ng dbus hal networkmanager alsa crond)
```

Agora, instale o KDM (lembrando que se você tem outro gerenciador de login já instalado, talvez não seja necessária esta instalação

```
# pacman -S kdm
```

E então, adicione o kdm também no */etc/rc.conf*:

```
DAEMONS=(syslog-ng dbus hal networkmanager alsa crond kdm)
```

Ok! Agora você tem um novo e lindo ambiente KDE rodando e funcional. Configure e estilize-o a vontade para ficar do jeito que você gosta!

7.4. Como instalar um ambiente XFCE

A instalação do Xfce também é algo bem simples. O Xfce é um ambiente bem enxuto e rápido, além de pequeno em tamanho de pacotes. Com ele podemos ter uma interface gráfica muito bonita e funcional em micros mais antigos e/ou com pouco poder de processamento.

- Instalando a base

```
# pacman -S xfce4
```

- Instalando alguns itens extras como plugins para o painel

```
# pacman -S xfce4-goodies
```

- Instalando o necessário para funcionar o som

```
# pacman -S gstreamer0.10-base-plugins
```

- Instalando os ícones padrão do menu

```
# pacman -S gnome-icon-theme
```

ou

```
# pacman -S tango-icon-theme
```

- Instalando o daemons necessários:

```
# pacman -S dbus
```

Adicione **dbus** em *DAEMONS* no */etc/rc.conf*

```
# pacman -S gamin
```

Agora, com seu usuário logado, teste o ambiente:

```
$ sudo /etc/rc.d/dbus start
```

```
$ startxfce4
```

Deu tudo certo ? Beleza, então vamos configurar o seu *~/.xinitrc*

Se você ainda não criou este arquivo, pode copiar o padrão do sistema: `cp /etc/skel/.xinitrc ~/`

Agora edite-o (*~/.xinitrc*) e adicione a linha a seguir:

```
exec ck-launch-session startxfce4
```

Ok, seu "startX" jah funciona. Mas agora, vamos instalar o login manager SLiM, ele utiliza o seu já configurado *~/.xinitrc*:

```
# pacman -S slim slim-themes archlinux-themes-slim
```

Veja/altere as configurações do slim em */etc/slim.conf*

Então adicione o "slim" no final do item *DAEMONS* no arquivo */etc/rc.conf*.

Pronto! Temos agora um ambiente Xfce leve e funcional.



Login manager

Como gerenciador de login para o Xfce você pode utilizar o GDM também, porém, aconselho a utilizar o SLiM que é mais prático e rápido.

7.5. Como instalar o SOM

Primeiro adicione seu usuário no grupo sound:

Instalação e configuração de programas

```
# gpasswd -a yourusername audio
```

- Vamos ver como instalar e configurar do ambiente de SOM PulseAudio.

Comece pelo básico:

```
# pacman -S pulseaudio
```

Instale os front-ends (GTK):

```
# pacman -S paprefs pavucontrol
```

Ele necessita do dbus para funcionar, então (instale e) inicie-o :

```
# /etc/rc.d/dbus start
```

Feito isto, para o ambiente não gráfico inicie o pulse assim:

```
$ pulseaudio --start
```

Ou em ambiente gráfico, assim:

```
$ start-pulseaudio-x11
```

E para os programas que não suportam o PulseAudio, devemos instalar o backend Alsa:

```
# pacman -S pulseaudio-alsa
```

Este pacote já contém a configuração necessária para fazer o Pulse funcionar com o Alsa, e se você está em um ambiente 64 bits, e quer ter som para os programas 32 bits, tenha certeza de ter instalado também as libs *lib32-libpulse* e *lib32-alsa-plugins*.

Para funcionar o gstreamer com o pulse, configure-o executando o comando *gstreamer-properties* e selecionando PulseAudio em ambas as configurações: Entrada e Saída de Áudio.

Para o Gnome tem ainda o *pulseaudio-mixer-applet*, que é um applet de controle do Pulse. O PulseAudio não é compatível com o KDE3, e para utilizar com o KDE4 faça conforme descrito nesta página: <http://www.pulseaudio.org/wiki/KDE> , e tem ainda um plasmoid para controle do Pulse: *kdeplasma-addons-applets-veromix*.

- Vamos ver como instalar e configurar o ambiente de som ALSA (Advanced Linux Sound Architecture)

O ALSA já é incluído no kernel padrão do Arch Linux, não sendo necessária sua instalação.

Então, instale as ferramentas

```
# pacman -S alsa-utils alsa-plugins
```

Agora, para o Gnome instale o *gnome-alsamixer*.

Capítulo 8. Como compilar o kernel do Jeito Arch ¹

Vamos admitir aqui que você utiliza e quer alterar o kernel padrão do Arch e já tem o ambiente do ABS configurado, conforme respectivo item deste livro - Seção 3.4, “Compilando através do ABS”.

Copie os arquivos para seu diretório de trabalho:

```
cp -R /var/abs/core/kernel26/* /path/to/my/abs/kernel26/
```

- Modifique o PKGBUILD conforme suas necessidades.
 - Descomente a linha atual do *pkgname* e descomente a linha abaixo, onde diz "kernel26-custom", trocando o "-custom" pelo que lhe convier.
 - Claro que você quer alterar as configurações do kernel, então descomente a linha do `make [menuconfig|nconfig|xconfig|etc]` que achar mais apropriada.

Agora compile o kernel com o comando *makepkg*. Pode ir tomar um café que este processo demora bastante. Demorar uma hora não é raro acontecer. Depois de terminada a instalação, instale seu novo kernel

```
# pacman -U kernel26-custom_i686.tar.pkg.xz
```

Então atualize o seu bootloader Grub e tudo acabado!

PS: Não esqueça que tem vários kernels diferenciados no AUR antes de meter a mão na massa, heim ! hehe (Veja o item Seção 8.1, “Kernels diferenciados” deste livro).

8.1. Kernels diferenciados

No AUR, estão disponíveis também alguns outros kernels com patch que ajudam em performance e/ou segurança. Abaixo falo um pouco sobre eles:

Con Kolivas (-ck)

Este kernel contém os patches criados por Con Kolivas ¹. Estes patches agregam melhorias no escalonador.

Real Time (-rt)

Este kernel contém patches para atividades de tempo real.

-grsecurity

Este kernel contém patches de segurança

Tiny Patches

¹Con Kolivas [<http://kernel.kolivas.org/>]

Estes patches são direcionados a melhorar o uso de memória e diminuir o footprint de disco. Direcionado a desenvolvedores de sistemas embarcados e/ou máquina com menos poder, como os antigos 386.

-fallen

Este é um agrupamento de vários patches: -ck , fbsplash, reiser4, suspend2, latest 2.6.x.y patch, ACPI DSDT, squashfs, e Genpatches

8.2. Dicas de Performance

O Arch Linux já é por padrão uma distribuição que trabalha com ótima performance, tem todos os seus pacotes compilados e voltados para arquitetura igual ou superior a i686, ou seja, processadores como Pentium Pro, Pentium II/III/Celeron, Pentium 4 e Athlon/Duron, não funcionando com processadores mais antigos como os 486 e 386, e os K6 que também apresentam alguns problemas. Estes processadores i686 processam duas instruções por vez, diferentemente destes mais antigos, que só processam uma instrução por vez.

Fora este detalhe da compilação, o Arch mantém uma vida simplista e simplificada, onde dois detalhes saltam aos olhos quando falamos disto: - O Arch não instala nada que você não saiba: Você sabe TUDO que roda no seu Arch, ele não vem com nada instalado ou rodando sem que não tenha sido você mesmo que tenha instalado. - Configuração centralizada: este também é um ponto bastante importante pois agiliza a configuração do ambiente, onde não necessitamos ficar alterando vários arquivos para configurar a distribuição. Items como rede, linguagem, serviços que sobem no boot e vários outros são configurados no arquivo `´/etc/rc.conf´`.

Mas mesmo assim há como manter seu ambiente de uma forma que ele trabalhe mais e mais rápido ? Abaixo listo algumas dicas para que você possa se guiar e aplicar aquelas que mais se encaixam com seu perfil de utilizador do sistema:

- Ambiente gráfico: escolha um ambiente gráfico mais leve e que utilize menos memória caso necessite, podemos citar aqui o Lxde e o Xfce por exemplo.
- Sistema de arquivos: para o linux existem vários sistemas de arquivos disponíveis, e cada um com suas características, e como podemos determinar como nosso disco vai ser particionado e onde cada partição será montada podemos fazer um sistema completamente configurado para performance, veja:
 - EXT4 - Tem excelente desempenho no geral, é confiável, mas ainda mantém alguns pequenos problemas com alguns bancos de dados, tem desfragmentação online e suporte a grandes volumes e número de diretórios.
 - ReiserFS - Ótima performance para arquivos pequenos, bem indicado para o `´/var´`.
 - XFS - Ótima performance para grandes arquivos, e baixa performance com arquivos pequenos. Ótima performance com os comandos de formatação, montagem e o fsck e utiliza poucos recursos de CPU.
 - Btrfs - Ainda em desenvolvimento, porém já constatado que tem melhor desempenho que o EXT4, e muitas features para boa performance. Podes utilizar para fazer testes.

- Conheça seu sistema
 - Veja a memória utilizada e livre com o comando `free -m`
 - Como está seu disco? Com o comando `hdparm -t /dev/harddrive` você verá o tempo exato de leituras do seu HD, não é exatamente um benchmark, mas são consideradas boas no mínimo de 40 MB/s.
 - Veja o que mais consome CPU com o comando `top` ou um mais visualmente agradável, o `htop`. De repente podem haver programas ou daemons rodando sem necessidade.
 - Tens uma placa 3D ? Veja se está configurado corretamente, `glxinfo | grep direct`, se não estiver instale os drivers da sua placa de vídeo para não denegrir a performance visual do ambiente.
- Teste e utilize aplicativos que são mais leves, e as vezes mais simples. Muitas vezes os aplicativos mais simples fazem tudo que os seus equivalentes mais pesados fazem, então sempre teste as várias opções até chegar ao que mais lhe agrada.
- Utilize um kernel mais performático. No AUR, como vimos em outro tópico deste livro (Seção 8.1, “Kernels diferenciados”), temos várias opções de kernels mais performáticos do que o padrão. Leia sobre eles e veja o que mais se encaixa em suas tarefas.
- Veynice é um aplicativo que ajuda na execução das tarefas no processador, ajustando de forma inteligente a prioridade de cada processo.
- Monte o diretório `/tmp` na memória RAM ao invés do disco, isto vai ajudar muito, porém vai comer parte da sua memória. É muito bom para quem tem bastante memória disponível.
- Diretivas de compilação: Ajuste corretamente as configurações de compilação específicas para seu processador, isto ajuda muito na hora de utilizar um programa compilado localmente.
- Se você compila bastante programas, utilize a ferramenta Ccache.
- Pacman : Ajuste o pacman para trabalhar melhor, tem várias dicas sobre isto como:
 - `pacman-optimize`: otimização da base local do pacman
 - `pacman-cage`: move a base do pacman para um local de disco onde se tem leitura contínua
 - melhore o tempo de downloads utilizando o Powerpill ou configurando a opção `xferCommand` no arquivo `/etc/pacman.conf` para utilizar o `aria2c` ou `axel`
 - Atualize a lista de mirrors do pacman para priorizar os mais rápidos em resposta utilizando o `rankmirror`
- E ainda, caso você necessite de um sistema realmente performático e quer ir mais a fundo nas configurações, você pode utilizar as ferramentas de benchmark citadas no wiki: <https://wiki.archlinux.org/index.php/Benchmarking> .

Considerações Finais

A pesquisa para a criação deste livro foi extensa. Sendo que encontrei a maioria das informações que me faltavam no wiki oficial, porém, várias dicas e informações adicionais eu encontrei também em alguns blogs e principalmente no fórum brasileiro, além da Wikipedia.

Desde já, agradeço a todos por terem disponibilizado tais informações e tutoriais, que com certeza, engrandecem a comunidade.

Locais de pesquisa

Sites

<http://www.archlinux.org/>

<http://www.archlinux-br.org/>

<http://wiki.archlinux-br.org/>

<http://wiki.archlinux.org/>

<http://pt.wikipedia.org/>

<http://memphyx.wordpress.com/>

<http://www.guiadopc.com.br/>

<http://www.guiadohardware.net/>

<http://jpl.com.br/art01.php>

<http://www.smithux.com/>