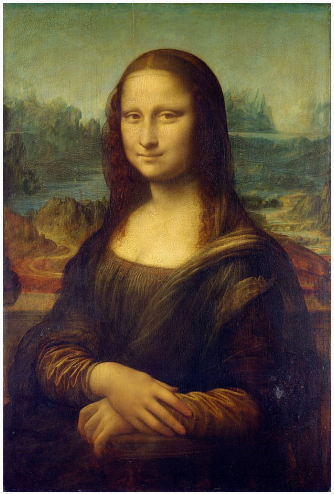
Uma viagem ao Kernel Linux

Rodrigo Siqueira
rodrigosiqueiramelo@gmail.com
http://siqueira.tech

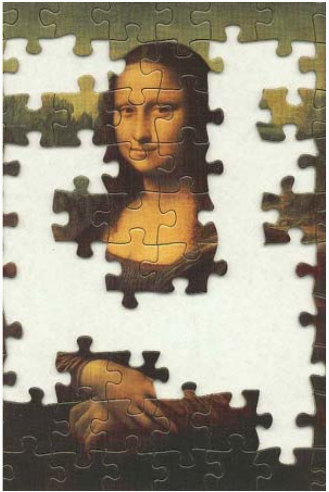
September 2, 2018

Parte 1: Viagem ao centro do Linux

Visão geral da comunidade Kernel Linux



Visão geral da comunidade Kernel Linux



Alguns dos subsistemas

Exemplos de subsistemas

alsa-devel, autofs, backports, ceph-devel, cgroups, cpufreq, dash, dccp, devicetree-compiler, devicetree-spec, devicetree, dmaengine, dwarves, ecryptfs, fio, fstests, initramfs, irda-users, kernel-janitors, kernel-packagers, kernel-testers, keyrings, kvm-commits, kvm-ia64, kvm-ppc, kvm, lartc, libzbc, linux-8086, linux-acpi, linux-admin, linux-alpha, linux-api, linux-apps, linux-arch, linux-arm-msm, linux-assembly, linux-bbs, linux-bcache, linux-block, linux-bluetooth, linux-btrace, linux-btrfs, linux-c-programming, linux-can, linux-cifs, linux-clk, linux-config, linux-console, linux-coverity, linux-crypto, linux-diald, linux-doc, linux-edac, linux-efi, linux-embedded, linux-ext4, linux-fbdev, linux-fido, linux-fpga, linux-fscrypt, linux-fsdevel, linux-fsf, linux-ftp, linux-gcc, linux-gpio, linux-hams, linux-hexagon, linux-hotplug, linux-hwmon, linux-i2c, linux-ia64, linux-ibcs2, linux-ide, linux-IIO, LInux-input, linux-integrity, linux-ipx, linux-isdn, linux-japanese, linux-kbuild, linux-kernel-announce, linux-kernel-announce.posters, linux-kernel, linux-kselftest, linux-laptop, linux-leds, linux-linuxss, linux-lugnuts, linux-m68k-cvscommit, linux-m68k, linux-man, linux-mca, linux-media, linux-metag, linux-mmc, linux-modules, linux-msdos-devel...

Os Mantenedores

Exemplos de subsistemas

Em software livre, um mantenedor de software ou mantenedor de pacotes é geralmente uma ou mais pessoas que criam código-fonte em um pacote binário para distribuição, fazem commits de patches ou organizam código em um repositório.

- O arquivo MAINTAINERS
- `scripts/get_maintainer.pl --separator , --nokeywords --nogit --nogit-fallback --norolestats <FILE/DIR>`

Linux Foundation e Repositórios

- `https://www.linuxfoundation.org/`
- `https://git.kernel.org/pub/scm/linux/kernel/git/`
- Algumas organizações tem seus próprios repositórios oficiais

Uma breve história do processo de desenvolvimento

1. Escrever o código
2. Testar o código
3. Verificar o estilo de código
4. Escrever uma boa menssagem de commit
5. Encontrar os maintainers e a lista de email correta para enviar o patch
6. Preparar o patch utilizando `git format-patch`
7. Enviar o patch
8. Esperar até 3 semanas por um feedback
9. Ao receber um feedback, aplicar as correções e reenviar o patch

Links importantes: leitura obrigatória

- Sobre o processo de desenvolvimento:
  `https://www.kernel.org/doc/html/v4.15/process/2.Process.html`
- Sobre o primeiro patch: `https://kernelnewbies.org/FirstKernelPatch`

O estilo de código

Importantes

- https://www.kernel.org/doc/html/v4.10/process/coding-style.html
- perl scripts/checkpatch.pl --terse --no-tree --color=always --codespell -strict --file <FILE>

Emails

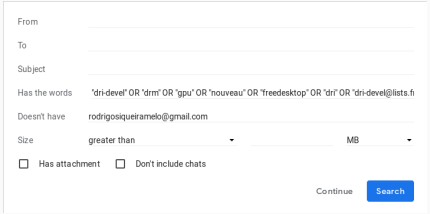
Passo 1: Configurar o neomutt

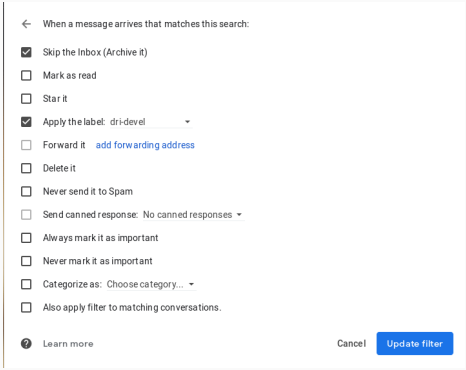
- Tutorial sobre mutt: `http://stevelosh.com/blog/2012/10/the-homely-mutt/`
  - Leia este tutorial para ter noções básicas da configuração, não se apegue aos detalhes
  - Recomendo pular toda parte de configuração offlineimapp
- Arquivo para o mutt pré-prontos:
  `https://github.com/rodrigosiqueira/myConfigFiles/tree/master/roles/neomutt/files/mutt`

Passo 2: Se inscrevendo em uma lista e criando filtros

- Utilizar uma conta gmail por questões de facilidade
- Criar um labels especificas (e.g., linux-kernel, dri-devel, etc)
- Configurar um filtro para redirecionar emails da lista para o label especifico

Emails: Exemplo de configuração de filtro





IRC

- A forma mais rápida de obter feedback da comunidade
- Cada subsistema tem a sua particularidade, logo, alguns estão no IRC e outros não

O Kernel Linux

Compilar o Kernel, por qual motivo você faria isto?

O Kernel Linux

Compilar o Kernel, por qual motivo você faria isto?

- Utilizando repositórios oficiais
- `git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git`

Uma visão geral dos arquivos

- arch
- Documentation
- drivers
- fs
- include
- init
- ipc
- Kbuild
- Kconfig
- kernel
- lib
- LICENSES
- MAINTAINERS
- Makefile
- mm
- net
- README
- scripts
- tools

O todo poderoso `.config`

O que é o `.config`

É o arquivo que diz o que será compilado

- Como obter um `.config` "funcional":
  1. `zcat /proc/config.gz > .config`
  2. `cp /boot/config-`uname -r` .config`
- Uma forma de reduzir a quantidade de arquivos compilados:
  `make localmodconfig`
- Navengando no menu de configurações
  `make nconfig`

Finalmente... compilando o Kernel

```
make -j16 && make modules
```

Instalando

ATENÇÃO: Os passos aqui descritos podem variar em cada distro

- `sudo make modules_install`
- `sudo make install`
- `sudo grub-mkconfig -o /boot/grub/grub.cfg`

Parte 2: Navegando no Kernel Linux com kworkflow

O que o kworkflow?

kworkflow (kw)

É um conjunto de scripts agrupados que tem por objetivo facilitar o ciclo de desenvolvimento no kernel Linux

- Kworkflow ou kw
- Simples de instalar

Visão geral do kw

- Instação: ./setup.sh -i

Visão geral do kw

- Instação: ./setup.sh -i
- Maintainers: kw maintainers <FILE/DIR>

Visão geral do kw

- Instação: ./setup.sh -i
- Maintainers: kw maintainers <FILE/DIR>
- Code style: kw codestyle <FILE/DIR>

Visão geral do kw

- Instação: ./setup.sh -i
- Maintainers: kw maintainers <FILE/DIR>
- Code style: kw codestyle <FILE/DIR>
- Preparar vm: kw prepare

Visão geral do kw

- Instação: ./setup.sh -i
- Maintainers: kw maintainers <FILE/DIR>
- Code style: kw codestyle <FILE/DIR>
- Preparar vm: kw prepare
- Compilar: kw build

Visão geral do kw

- Instação: ./setup.sh -i
- Maintainers: kw maintainers <FILE/DIR>
- Code style: kw codestyle <FILE/DIR>
- Preparar vm: kw prepare
- Compilar: kw build
- Instalar módulos: kw install

Visão geral do kw

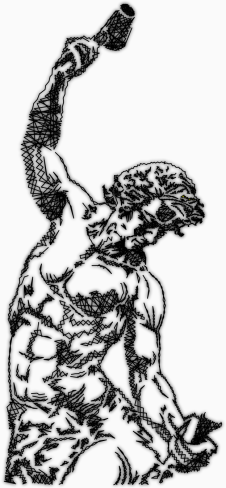
- Instação: ./setup.sh -i
- Maintainers: kw maintainers <FILE/DIR>
- Code style: kw codestyle <FILE/DIR>
- Preparar vm: kw prepare
- Compilar: kw build
- Instalar módulos: kw install
- Compilar e instalar: kw bi

Ajuda

Figure 1: `https://github.com/rodrigosiqueira/kworkflow.git`

Parte 3: Navegando no em você mesmo

Boa parte do trabalho parte de você



Você não está sozinho...

Uma viagem ao Kernel Linux

Rodrigo Siqueira
rodrigosiqueiramelo@gmail.com
http://siqueira.tech

September 2, 2018