

# systemd: repensando a inicialização

Lucas De Marchi

ProFUSION

6 de agosto de 2011



# Quem sou eu?

Engenheiro de software na ProFUSION embedded systems;

Participante de alguns grande projetos open source: Linux Kernel, BlueZ, oFono, ConnMan, WebKit, EFL;

Criador de outros poucos: dietsplash, codespell, trissa 3D;

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

IRC: demarchi



# Introdução - O que é systemd?

systemd é um sistema de **init** para o sistema operacional Linux.

O que exatamente é um “sistema de init”?

# Introdução - O que é systemd um sistema de init?

systemd é um sistema de **init** para o sistema operacional Linux.

O que exatamente é um “sistema de init”?

- 1 Primeiro processo executado pelo kernel
- 2 Montar partições
- 3 Processo responsável por trazer todos os serviços durante o boot

# Introdução - O que é `systemd` um sistema de `init`?

Um pouco mais que isso:

- 1 Configurar `hostname` / `domainname`
- 2 Conferir se partições contém erros
- 3 Carregar módulos do kernel necessários
- 4 Gerenciar logins
- 5 Configurar console / mapa de teclado / locale
- 6 Configurar relógio do sistema / `timezone`
- 7 Desligamento: último processo a morrer
- 8 Universal child reaper

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux,*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts.*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities,*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services,*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons,*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, keeps track of processes using Linux cgroups,*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, keeps track of processes using Linux cgroups, supports snapshotting and restoring of the system state,*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, keeps track of processes using Linux cgroups, supports snapshotting and restoring of the system state, maintains mount and automount points and implements an elaborate transactional dependency-based service control logic.*

# Introdução - O que é systemd?

**systemd**, um pouco mais que isso?

*“systemd is a system and session manager for Linux, compatible with SysV and LSB init scripts. systemd provides aggressive parallelization capabilities, uses socket and D-Bus activation for starting services, offers on-demand starting of daemons, keeps track of processes using Linux cgroups, supports snapshotting and restoring of the system state, maintains mount and automount points and implements an elaborate transactional dependency-based service control logic. It can work as a drop-in replacement for sysvinit.”*

# Inicialização do SO

- First-stage bootloader (X-loader)
  - ↪ Second-stage bootloader (u-boot)
    - ↪ Kernel (Linux)
      - ↪ Init (systemd)

ou

- First-stage bootloader (X-loader)
  - ↪ Second-stage bootloader (u-boot)
    - ↪ Kernel (Linux)
      - ↪ Initram (dracut)
        - ↪ Init (systemd)

# Outros sistemas de init

- SysV init:
  - ▶ Mais famoso sistema de init em sistemas \*nix;
  - ▶ Runlevels
  - ▶ Configuração: /etc/inittab, /etc/rc.d/\*
  - ▶ Ordem determinada pelo nome do arquivo
  - ▶ Alta dependência de shell scripts: um sistema geralmente boota por completo usando 3000+ processos
  - ▶ Projeto antigo, bem estável
- BSD: um pouco diferente do SysV - não tem o conceito de runlevels, ordem é determinada por tags em cada script de init

# Outros sistemas de init

- SysV init:

- ▶ Mais famoso sistema de init em sistemas \*nix;
- ▶ Runlevels
- ▶ Configuração: /etc/inittab, /etc/rc.d/\*
- ▶ Ordem determinada pelo nome do arquivo
- ▶ Alta dependência de shell scripts: um sistema geralmente boota por completo usando 3000+ processos
- ▶ Projeto antigo, bem estável

Algum problema com esse sistema?

# Outros sistemas de init

- upstart:

- ▶ Idealizado por Scott J. Remnant para o Ubuntu (circa 2007)
- ▶ Lê arquivos de configuração em `/etc/init/*`
- ▶ Baseado em eventos: “start on”, “stop on”, “initctl emit”
- ▶ Diminui dependência de shell scripts, mas continua sendo bastante usado

Algum problema com esse sistema?

# Outros sistemas de init

Sistemas menos conhecidos/usados:

- **OpenRC**: sistema usado no Gentoo
- **initng**: Gentoo init wanna be
- **runit**

Sistemas embarcados:

- **ltib**
- **upstart**
- **Android**
  - ▶ Design similar ao do upstart
  - ▶ Baseado em eventos
  - ▶ Eventos são lidos de `/init.rc` e `/init.plataforma.rc`
- **home made**

# Systemd - sobre

## Repensando o init: systemd

- Criado no final de 2009, começo de 2010
- Autores e mantenedores: Lennart Poettering e Kay Sievers
- Utiliza recursos exclusivos do Linux: cgroups, devtmpfs, signalfd, epoll, notificação de swap/mount point, etc. Não há preocupação em ser portátil para outros sistemas.
- Utilização excessiva de scripts é prejudicial: lentidão na execução e baixa padronização
- Execução sequencial é a mais simples e burra possível
- Atualmente a criação de serviços (daemons) é difícil, sujeita a erros e não-consistente.
- Configuração: cada distribuição faz como quiser (Exemplos: hostname, relógio, mapa de teclas etc)
- Preocupa-se em como iniciar o serviço da forma mais simples, mas não como mantê-lo rodando

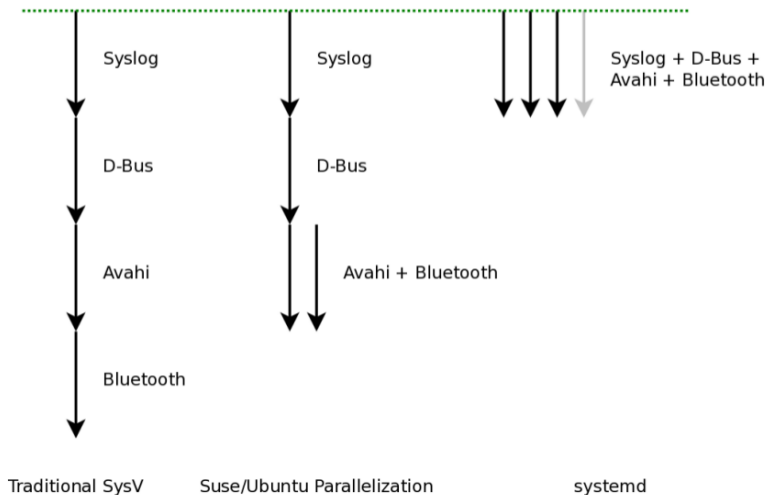
# Systemd - características

- Dependências implícitas/explicitas
- Alta paralelização do boot
- Diversas formas de iniciar um serviço
- Inicialização sob demanda de serviço
- Baixa utilização de shell scripts
- Facilita a criação de serviços: parte da tarefa que era executada nele passa a ser executada pelo próprio systemd

Fortemente dependente de algumas características/programas do Linux: cgroups, udev, d-bus, dracut (quando há initram)

Bootsplash? Não! Plymouth (desktop) ou dietsplash (sistemas embarcados)

# Systemd - comparação com outros inits



# Systemd - Objetivos maiores

- Padronização
- Velocidade de boot: em alguns casos já é possível bootar em  $< 1s$
- Utilizar toda a “largura de banda” disponível
- Boot sem shell scripts
  - ▶ Tarefas de sistema
  - ▶ Arquivos de configuração de inicialização de serviços
- Integração
- Modularização
- Paralelização
- Manutenção do estado dos serviços

# Systemd - sistemas embarcados?

Mas esse sistema de init se adequa a sistemas embarcados?

**Depende!** Quer um sistema

- Padronizado,
- utilizando toda a banda disponível,
- veloz,
- sem shell scripts e suas gambiarras,
- bem integrado,
- modular,
- com serviços inicializados em paralelo,
- tendo uma baby-sitter pra monitorar os serviços

# Systemd - sistemas embarcados?

Mais seriamente: sim! Com exceção de alguns serviços desnecessários na maior parte dos cenários de sistemas embarcados:

- hostnamed
- logind / pam\_systemd
- locale
- timedated

Suporte atual para **Ångstrom, Yocto e Gentoo.**

# Systemd - ativação por socket

- Kernel ordena e bufferiza requisições
- Dependência implícita
- Necessita suporte por parte do daemon

**Vantagem: serviços que utilizam um outro podem ser iniciado antes. Exemplo: syslog**

# Systemd - ativação por D-Bus

- systemd registra determinado nome barramento
- Serviço é iniciado quando alguma conexão é realizada
- Necessita suporte por parte do daemon

**Vantagem: serviços que utilizam um outro podem ser iniciados antes. Exemplo: Bluetooth**

# Systemd - ativação por caminho (path)

- systemd monitora determinado arquivo
- Serviço é iniciado quando evento acontece nesse arquivo

**Vantagem: serviço pode ser iniciado somente quando necessário.**

**Exemplo: cups**

# Systemd - ativação sob demanda

- socket
- D-Bus
- path

**Vantagem: serviços que utilizam um outro podem ser iniciados antes e somente quando necessários.**

# Systemd - babysitter

Systemd monitora qual a política de (re)-inicialização:

- nunca
- sempre
- só quando termina com sucesso
- só quando termina com falha

## Systemd - babysitter

Diversas outras políticas podem ser setadas para um determinado serviço:

*Environment, resource limits, working directory, chroot(), umask, OOM adjustment, nice level, IO priority and class, CPU scheduler priority and policy/reset-on-fork, CPU affinity, timer slack, stdio to syslog/tty/null/kmsg, uid, gid, supplementary groups, file system namespaces (r/o file systems, inaccessible systems, mount propagation, private /tmp), capabilities (inherited set, bounding set, secure bits), ...*

Veja: `man systemd.service`

# Systemd - configuração

Arquivos de configuração do systemd do sistema:

`/lib/systemd`

Arquivos de configuração do systemd do usuário e/ou  
ativados/desativados: `/etc/systemd`

# Systemd - comandos úteis

- **systemctl:** inicia, termina, habilita serviços
- **systemd-analyze:** analisa o tempo de boot para poder ser otimizado
- **systemd-cgls:** saída do tipo **ps tree** mas com as verdadeiras relações de pai e filho dos processos
- **systemd:** emite comandos diretamente para o PID 1