

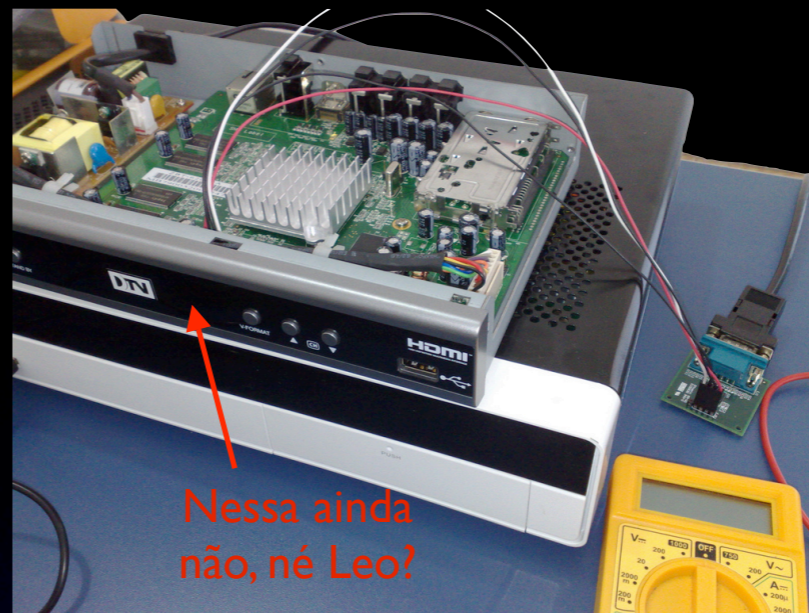
Python em todos os lugares

Oswaldo Santana Neto
osantana@gmail.com



Everywhere





Onde estamos?

ARM/XScale, x86 e **Mips**



Como avançar?



CPython

Cross-compiling

Usar um compilador em uma plataforma A
para gerar código para uma plataforma B

Cross-compiling,
autotools e make



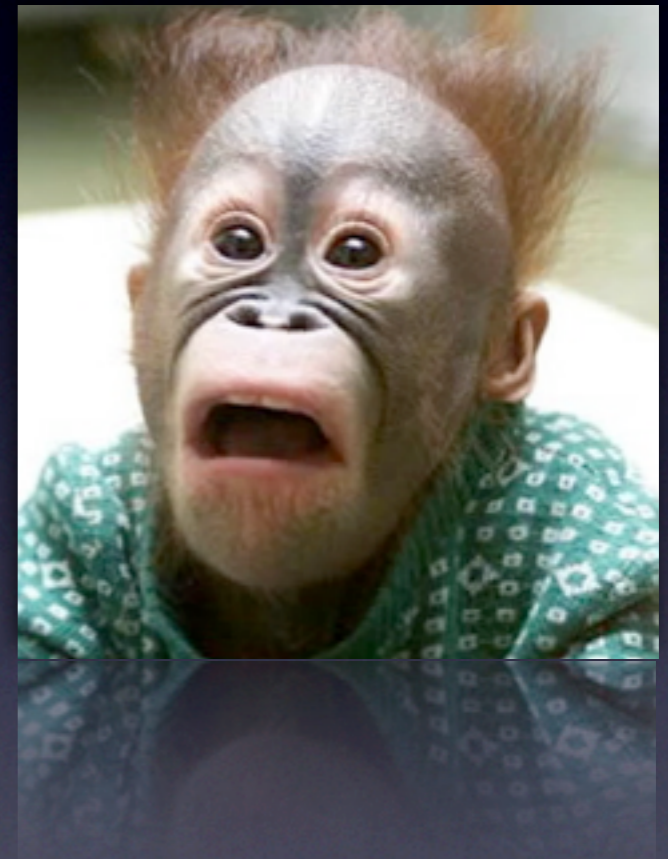
triveos 

distutils

distutils + cross-compiling



distutils + cross-compiling =



Cross-compiling

- `./configure --host=arquitetura-alvo`
- `make`
- `make install DESTDIR=/rootdir-alvo`

Cross-compiling Python

○ jeito “patch-patch-compila-guarda-compila”

Cross-compiling Python

- Prós

- É o jeito “certo”, multiplataforma, de se fazer isso.
- O patch em questão já foi discutido e só não foi aceito por falta de alguém para endossar o patch.

- Contras

- Dá um trabalhão.
- É necessário muitas intervenções no Setup.py e corre-se um risco grande de usar bibliotecas da plataforma incorreta.

Scratchbox (ad-hoc)

Host - x86 - Linux

binfmt

ELF ARM = /usr/bin/qemu-arm-static

Target - ARM - Linux

chroot

cross-
compiler

Deve ser
instalado para
se comportar
como
compilador
nativo

qemu-
arm-static

Deve ser
estático para
não precisar
do linker
dinâmico x86

Solução usada na plataforma Maemo que fornece um ambiente “semi-emulado” para desenvolvimento.

Cross-compiling Python

○ jeito “./configure; make; make install”

Cross-compiling Python

- Prós

- Não é necessário modificar nada no Python
- O ambiente serve para compilar qualquer tipo de aplicação (mesmo as que não prevêm isso)

- Contras

- O ambiente precisa estar completo (ex. header files)
- O qemu geralmente precisa ser “patchado” e pode congelar com aplicações que usam *threads*

Ainda dá tempo?

3 slides pra falar de ctypes?

Bibliotecas Proprietárias e Python

- Fazer uma extension em C leva muito tempo.
- Algumas vezes os header files não estão disponíveis
- Entra em ação o módulo ctypes!
- Wrapper pra libffi

○ “Hello World” da ctypes

```
example_ctypes.py
from ctypes import *

libc = CDLL("/usr/lib/libc.dylib")

class TimeT(Structure):
    _fields_ = [
        ('tm_sec', c_int), ('tm_min', c_int), ('tm_hour', c_int),
        ('tm_mday', c_int), ('tm_mon', c_int), ('tm_year', c_int),
        ('tm_wday', c_int), ('tm_yday', c_int), ('tm_isdst', c_int),
        ('tm_zone', c_char_p), ('tm_gmtoff', c_long),
    ]

time = TimeT()
time.tm_sec = 0; time.tm_min = 7; time.tm_hour = 1
time.tm_mday = 4; time.tm_mon = 8; time.tm_year = 2008
time.tm_wday = 2; time.tm_yday = 248; time.tm_isdst = 0
time.tm_zone = "BRT"; time.tm_gmtoff = -10800

libc.printf("Hello World!\n")

libc.asctime.restype = c_char_p
libc.printf( libc.asctime(byref(time)) )
```

```
osvaldo-santanas-macbook:Documents osantana$ python example_ctypes.py
Hello World!
Tue Sep  4 01:07:00 3908
```

Line: 1 Column: 21 Python Soft Tabs: 4

... e muito mais.



Perguntas?

