python - easy deploing

Christian Kniep

Internation Center of Applied Technologies Bandung

2. August 2010

Table of content

Once upon a time...

- 'ugly' assembler had to be used (e.g. MIPS)

```
.data # define some varbiables
# the string we want printed
out:  .asciiz "Hello World"

.text # the program
main: li $v0, 4    # cmd-reg to cmd 4 ('print')
      la $a0, out  # set out as the 1st arg
      syscall      # execute the cmd
      li $v0, 10   # cmd-reg to cmd 10 ('exit')
      syscall      # execute the cmd
```

don’t get me wrong

- Assembler is the closest,directest way to program a CPU

$\Rightarrow$ so its the fastest code you can write

- you might just want to switch on the coffee-maker

$\Rightarrow$ you don’t care if it takes 2 nano- or 200 milliseconds

High Level Programming

- it doesn't mean you have to have a high level to program it
- it the opposite: you only have to handle abstracted commands the assembler code will be created for you
- checkout 'hello world' in C

```
#include <stdio.h>

main()
{
  printf("Hello World \n");
}
```

python-Programming

- python is even more high level then c
  - doesn't care what type you are using
  - the syntax is intuitive
  - its an interpreter language, so you don't have to compile it
  - it's build from scratch, so there is no(t much) historical payload

```
print "hello world"
```

whats this suppose to mean?

- you don't have to create an binary file
- the programm is evaluated linewise

$\Rightarrow$ so you are able to get a prompt

```
$ python
>>> print 'Hello World'
Hello World
```