SYS1 - Harmonie

$a_0 \sum_{n=1}^{\infty} a_n \cos(n2\pi\nu t) + b_n \sin(n2\pi\nu t)$

Jules Aubert

Sommaire

1. Bibliography
2. Makefiles
3. Shell
4. Vim
5. Brainteasers
6. Questions?

Sommaire

1. Bibliography
2. Makefiles
3. Shell
4. Vim
5. Brainteasers
6. Questions?

Sommaire

1. Bibliography
2. Makefiles
3. Shell
4. Vim
5. Brainteasers
6. Questions?

Sommaire

1. Bibliography
2. Makefiles
3. Shell
4. Vim
5. Brainteasers
6. Questions?

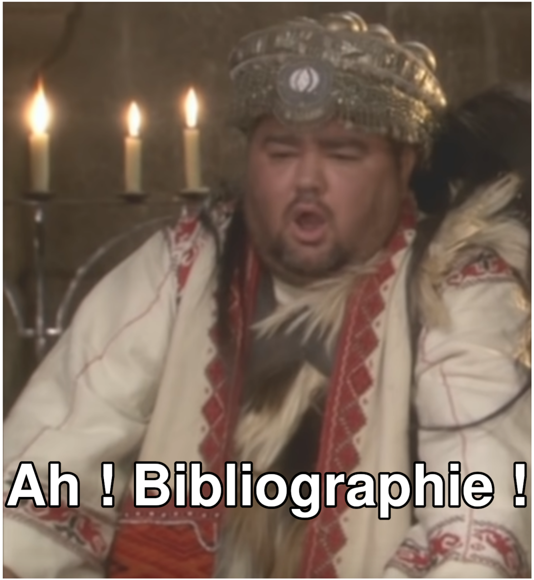
Sommaire

1. Bibliography
2. Makefiles
3. Shell
4. Vim
5. Brainteasers
6. Questions?

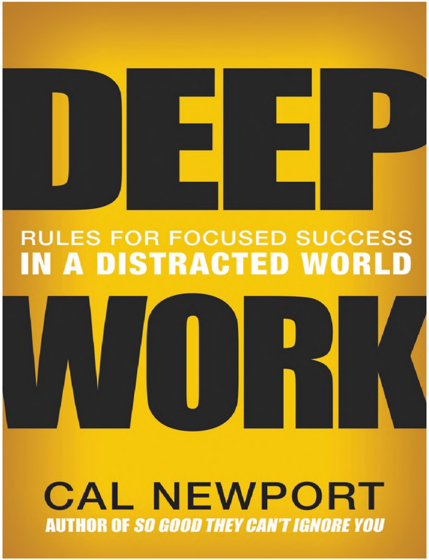
Sommaire

1. Bibliography
2. Makefiles
3. Shell
4. Vim
5. Brainteasers
6. Questions?

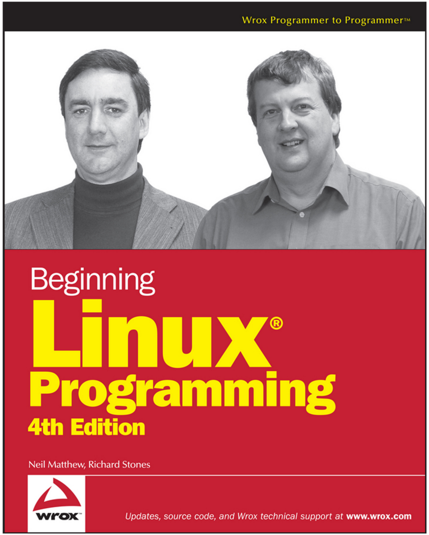
Bibliography



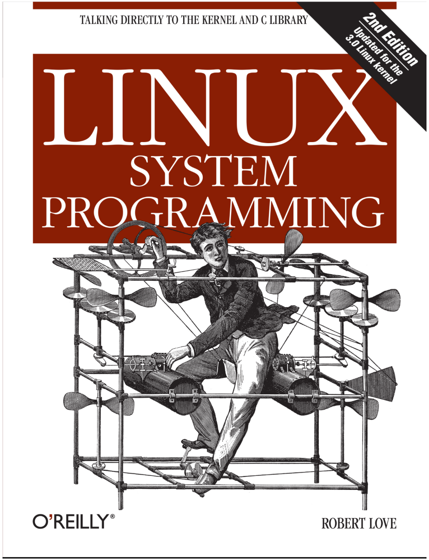
Deep Work


DEEP
RULES FOR FOCUSED SUCCESS
IN A DISTRACTED WORLD
WORK
CAL NEWPORT
AUTHOR OF SO GOOD THEY CAN'T IGNORE YOU

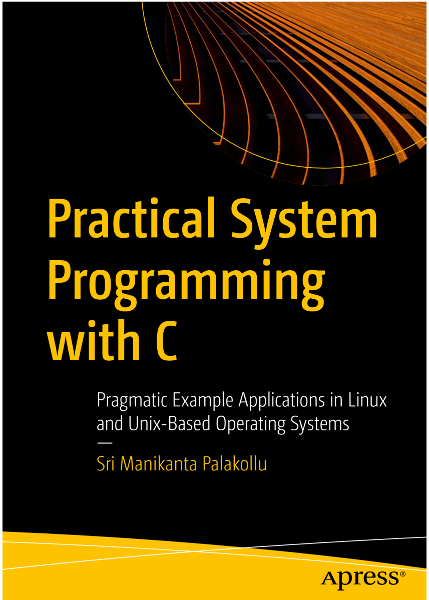
Beginning Linux Programming


Wrox Programmer to Programmer™
Beginning
Linux®
Programming
4th Edition
Neil Matthew, Richard Stones
wrox
Updates, source code, and Wrox technical support at www.wrox.com

Linux System Programming


TALKING DIRECTLY TO THE KERNEL AND C LIBRARY
2nd Edition
Updated for the 3.0 Linux kernel
LINUX
SYSTEM
PROGRAMMING
O'REILLY®
ROBERT LOVE

System Programming in C



Practical System Programming with C

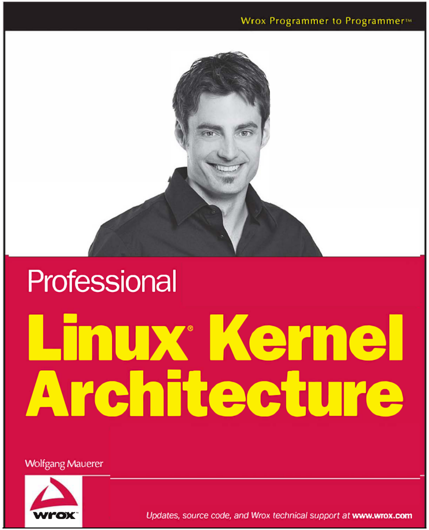
Pragmatic Example Applications in Linux and Unix-Based Operating Systems

—

Sri Manikanta Palakollu

Apress®

Linux Kernel Architecture



Wrox Programmer to Programmer™

Professional

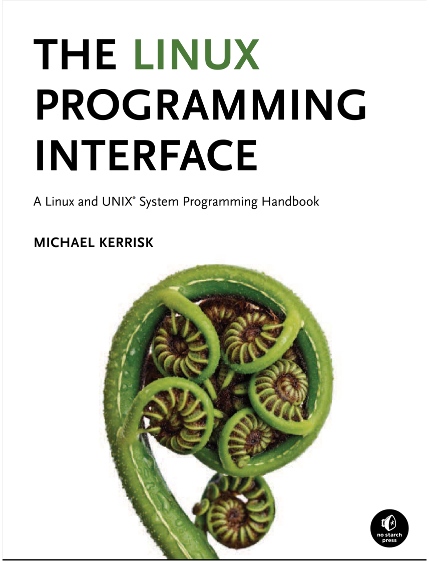
Linux® Kernel Architecture

Wolfgang Mauerer

wrox™

Updates, source code, and Wrox technical support at **www.wrox.com**

Linux Programming Interface



THE LINUX PROGRAMMING INTERFACE

A Linux and UNIX® System Programming Handbook

MICHAEL KERRISK

no starch press

Holy Grail

kernel.org

docs.kernel.org

OSDev

wiki.osdev.org

DEVSE

devse.wiki

Makefiles

It's demo time!

https://github.com/Gelules/makefiles

Shell - Builtins

- help
- builtin
- type
- which

Shell - test like a pro

```
man test
```

Shell - loop at the swimming pool, don't poop (lool)

- for with seq
- for without seq
- whiling a file

For with a seq

```
for i in $(seq 2 5 26)
do
    echo $i
done
```

It uses a subshell and invokes the seq(1) binary

For with pure shell

```
for i in {2..26..5}
do
    echo $i
done
```

Read a file line by line with while

```
while read line
do
    echo $line
done < the_file_to_read
```

Shell - Functions

https://github.com/Gelules/dotfiles

Undefined Behavior

Try it with sh, bash, zsh, ksh, csh, . . .

```
rm -rf toto
mkdir toto
cd toto
rmdir ../toto
cd .
pwd
echo $PWD
```

What just happened?

Brain teaser

I accidentaly execute

```
sudo chmod -x $(which chmod)
```

How can I retrieve the chmod execution bit?

Brain answer

I could use chmod(2)

Or use ld.so to load the binary in a memory zone that can execute code

```
sudo /lib64/ld-linux-x86-64.so.2 /usr/bin/chmod +x /usr/bin/chmod
```

sizeof *

I got a function pointer. The function takes 9 chars and 2 ints and returns a float.

What it the size of my pointer?

sizeof *

```c
float func(char c1, char c2, char c3,
           char c4, char c5, char c6,
           char c7, char c8, char c9,
           int i1, int i2)
{
    return (float)(c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8 + c9 + i1 + i2);
}

int main(void)
{
    float (*func_ptr)(char, char, char,
                      char, char, char,
                      char, char, char,
                      int, int) = func;

    return sizeof (func_ptr);
}
```

Vim

Chewie's vim conf Conférences 42sh 2019 - VIM / Emacs

TODO: speedrun the

- buffer
- split
- tab,
- autocompletion (text and files)
- compilation with the quickfix window
- terminal
- debugger

(**all vanilla**) to show interests

Black conf

I was supposed to do this in a bar with you all at the end of the Piscine... but nothing happened :'(

Talk to the BEABA asso about it ;)

I may consider doing black confs this year in class :)

?

Questions?