

Introducing

YULUG

by bandali

Outline

Outline

- Brief history of GNU/Linux

Outline

- Brief history of GNU/Linux
- Why use GNU/Linux

Outline

- Brief history of GNU/Linux
- Why use GNU/Linux
- Getting started with GNU/Linux

About me

About me

Work experiences

Summer 2013 - Fall 2014:

Software Developer @ Athlete Builder Inc.

Spring 2013:

Application Developer @ VONICAL Inc.

Winter 2013:

Mobile & Web Developer @ Hire Works Inc.

kelar.org/~bandali/cv.html

About me

Work experiences

- Tools: Emacs, i3, Git, Zsh, Vim
- Languages: Python, Haskell, Clojure, C, Java

Tools & languages

About you

Which OS do you use?

About you

Which OS do you use?

- Windows

About you

Which OS do you use?

- Windows
- OSX

About you

Which OS do you use?

- Windows
- OSX
- GNU/Linux

About you

Which OS do you use?

- Windows
- OSX
- GNU/Linux
- BSD's and other unices

Brief History of GNU/Linux

Unix

Unix

What

- "Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others." —Wikipedia (Unix)

Unix

What

- "Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others." —Wikipedia (Unix)
- The first portable OS (entirely written in C)

Unix

What

- "Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others." —Wikipedia (Unix)
- The first portable OS (entirely written in C)
- Key characteristic: Modular design

Unix

What

Philosophy

- "Write programs that do one thing and do it well."
- "Write programs to work together."
- "Write programs to handle text streams, because that is a universal interface."

–Douglas McIlroy (Inventor of the Unix pipeline)

Unix

What

- Some versions are proprietary

Philosophy

- Others are free/libre software

License

- Cost:
 - A \$20,000 license from Bell Labs
 - Educational license: \$150

GNU

In 1983, Richard Stallman announced the GNU project, an effort to create a "free" Unix-like OS.

GNU

Software freedoms

"A program is free software if the program's users have the four essential freedoms:

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this."

—What is free software? - GNU Project

GNU/Linux

- GNU missing a working kernel

GNU/Linux

- GNU missing a working kernel
- In 1991 Linus Torvalds released his hobby, the Linux kernel, as a replacement for the Minix kernel.

GNU/Linux

- GNU missing a working kernel
- In 1991 Linus Torvalds released his hobby, the Linux kernel, as a replacement for the Minix kernel.
- Licensed under GNU GPL (General Public License)

GNU + Linux = GNU/Linux

Why Use GNU/Linux

Freedom

Choices

Customizability



Desktop



Documents



Downloads



Dropbox



Music



Pictures



Public



Templates



Videos

+Ivan Petruk



Radioacti

onix & Lindse

RADIC
PENTATONIX



```
Code Pro";
```

118,1 47%

```
erving ./ on: http://0.0.0.0:8080
er
GMT] "GET /" "Mozilla/5.0 (X11; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0"
GMT] "GET /css/style.css" "Mozilla/5.0 (X11; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0"
GMT] "GET /js/scripts.js" "Mozilla/5.0 (X11; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0"
GMT] "GET /favicon.png" "Mozilla/5.0 (X11; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0"
GMT] "GET /2015/01/05/testing-css.html" "Mozilla/5.0 (X11; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0"
GMT] "GET /img/emptybar.jpg" "Mozilla/5.0 (X11; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0" /u/juhohei
```

Facebook

testing css | juhohei

0.0.0.0:8080/2015/01/05/testing-css.html

TESTING CSS

A paragraph. This paragraph has a [link](#), turquoise in color. If you hover it, it'll be blue. Here's some more text just to make the paragraph at least three rows in length.

Second paragraph with `inline code`. It'll be turquoise color with gray background. Next is a block of code. It should have black background with white text.

```
function hello (name) {
```



```
1 (require [clojure.set :as set])
2
3
4 (def board identity)
5
6 (def all-values #{1 2 3 4 5 6 7 8 9})
7
8 (defn value-at [board coord]
9   (get-in board coord))
10
11 (defn has-value? [board coord]
12   (if (zero? (value-at board coord)) false true))
13
14 (defn row-values [board coord]
15   (let [[row _] coord]
16     (set (get board row))))
17
18 (defn col-values [board coord]
19   (let [[_ col] coord]
20     (set (for [n (range 0 9)]
21            (value-at board [n col])))))
22
23 (defn coord-pairs [coords]
24   (for [row coords
25         col coords]
26     [row col]))
27
28 (defn block-coords [board coord]
29   (let [[x _] coord
30         [_ y] coord]
31     xlst (cond
32            (<= 0 x 2) 0
33            (<= 3 x 5) 3
34            :else 6)
35     ylst (cond
36            (<= 0 y 2) 0
37            (<= 3 y 5) 3
38            :else 6)
39     xrest (range xlst (+ xlst 3))
40     yrest (range ylst (+ ylst 3))]
41     (for [xlst xrest
42           ylst yrest]
43         [xlst ylst])))
44
```

```
el s6b, 29 dic 2012
o Canadian for this shit. I'm not even Canadian.
problems with epsdribble being a shit and not
```

```
restart x
```

```
e recompilations it worked tho
.xinitrc runs
put epsdribble in .xinitrc
```

```
en epsdribble
```

```
pi
```

```
ido ahora cosa dum
his is taken
do ahora cosa tapi
it
```

```
ays
o: 0.529 H: 6 12 8 5 12 4 4 7 4
```

```
##c
##programming
##archlinux
##vim
##/g/technology
##rice
```

```
next(cr)-
```

```
t wrapl(cr)-
```

```
ent! set paste<CR>*piset nopaste<CR>-
paste<cr> z-
```

```
visual mode when I mean to "y". Use "gu" for those rare occasions.-
enrik/dotfiles/blob/master/via/config/mappings.vim-
```

```
C -> CR -> <cr>-
```

```
silent !yctags >/dev/null 2>&1 &<cr>:redraw!<cr>-
```

```
synIDattr(synID(line(".",col(".",1),name") . "> trans<-
synIDattr(synID(line(".",col(".",0),name") . "> to<-
synIDattr(synIDtrans(synID(line(".",col(".",1),name") . ">)<CR>-
```

/u/hamsterslayer01

```
halia@polaris:~$ >> about -haliax
```

```
main          japan
blog          manga
lastfm        l-novel
git           releases
mail          eat
```

```
utils         forums
vidhost       4chan
books         archlinux
walls         reddit
imghost       jcafe24
```

```
beans         bits
unixporn      anisebytes
dotshare      bitstnoop
/g/technology rutracker
nixers        pirateboy
```

```
halia@polaris:~$ >>
```

Getting Started with GNU/Linux

YULUG

[<https://yulug.org>]