

# PB173 – Ovladače jádra – Linux

## I.

Jiří Slabý

ITI, Fakulta Informatiky

21. 9. 2010

- Semestr = 12 týdnů (28. 9. svátek)
- Cvičící
  - Vývoj jádra od r. 2005 (NetBSD, Linux)
  - Student FI
- Cíle cvičení
  - Nastínit trochu jiný model programování
  - Prohloubit znalosti vnitřností OS a HW
- Ukončení: k
  - Splnění úkolů (možno ve skupinkách)

Vše potřebné na: [https://minotaur.fi.muni.cz:8443/~xsvenda/docuwiki/doku.php?id=public:pb173:pb173\\_2010\\_systemlinux](https://minotaur.fi.muni.cz:8443/~xsvenda/docuwiki/doku.php?id=public:pb173:pb173_2010_systemlinux)

# S čím budeme pracovat?

- satyr01-satyr10
  - CentOS 5.5
  - Login/heslo: vyvoj/vyvoj
- GIT
  - Úvod do GITu dnes
  - Podrobněji: <http://book.git-scm.com/>
- Zdroje jádra
  - Použijeme předinstalované z RPM (`/usr/src/kernels/`)
  - <http://git.kernel.org/> (od 2.6.12)
  - `full-history-linux` GIT (od 0.01 do 2.6.26)
- QEMU
  - Ze začátku
- COMBO6X karty
  - Později; k PCI, I/O, na přerušení, DMA
  - <http://www.liberrouter.org/>

## Práce s GITem

- 1 Na aise ve svém ~ proveďte klon
  - `git clone git://decibel.fi.muni.cz/~xslaby/pb173`
- 2 Změňte cokoliv v souboru `sandbox/hello`
- 3 Zkontrolujte změny (`git diff --color`)
- 4 `git commit -a` (správný log: suma, volný řádek, zdůvodnění, volný řádek, podpis (-s); vzor `git.kernel.org`)
  - Git může chtít nastavit jméno+e-mail (instrukce jsou na stdout)
- 5 Smažte `sandbox/hello` (`git rm`)
- 6 `git commit -a`
- 7 Zkontrolujte log, zda obsahuje 2 změny (`git log --color`)
- 8 Vygenerujte 2 patche (`git format-patch -2`)

## Co s patchem dále?

- Poslat odpovídajícímu správci (`scripts/get_maintainer.pl`)
- Poslat na ML a vystavit celý GIT strom, aby někdo provedl „merge“
- Popis v [Documentation/SubmittingPatches](#)

## Vývojový proces jádra

- L. Torvalds má právo veta
- Záplatováním se opravuje i přidává nová funkcionality
- Ovladače mohou být zařazeny přes `drivers/staging`
  - Začištění, smazání nepotřebného a přesun ze staging
  - Správce Greg KH

**Stejným způsobem odevzdávání úkolů**

## Hlavní rozdíly

- Žádné *libc* (`printf`, `strlen`, `malloc`, ...), ani ostatní (`pthread`)
- Ne/oddělený *paměťový prostor*
- *Počáteční funkce* (`main`)
- Pád systému  $\Rightarrow$  pád všeho

- GNU C (x86 už jen gcc >= 4.x)
- *CodingStyle*
  - Kontrola: `scripts/checkpatch.pl` (ne 100%)
- Monolit (`vmlinux` → (b) zImage)
  - Ale moduly (standardní ELF: \*.ko)
- `module_init` (=main), `module_exit` (=on\_exit)

## Dokumentace

- `Documentation/`\*
- Generovaná tamtéž (podobná DocBook)
- Kód (<http://lxr.linux.no/>)

Demo: pomocí lxr najdete v jádře obecnou (v `lib/`) a optimalizovanou (v `arch/`) implementaci `strlen`

- 1 Spusťte QEMU (`./qemu-start /home/local/centos.img`)
- 2 `git clone`  
`git+ssh://xlogin@aisa.fi.muni.cz/home/xlogin/pb173`
- 3 Prozkoumejte adresář 01 z pb173 git repozitáře
  - Makefile, pb173.c
- 4 Do `init` funkce doplňte výpis „Hello World”
  - `printk, KERN_*`
  - Objeví se v `/proc/kmsg` a na konzoli
- 5 Přeložte a vložte do systému
  - `make, insmod`
- 6 Zkontrolujte výstup `dmesg`
- 7 `git commit -a`
- 8 Pošlete na aisu (`git push`)



- Omezený zásobník (4k-8k)
  - Žádná nebo malá rekurze
  - Jen pro malá data (`int`, malé `struct`, krátká pole, ...)
- Pracuje se se stránkami (na x86 4K-1G)
  - Nesouvislá paměť/halda
  - Omezená velikost alokace (fragmentace)

- 1 Po stránkách (makro `PAGE_SIZE`)
  - `__get_free_page`, `free_page`
  - `get_free_pages`, `free_pages` (řád)
  - Vraccioj adresu jako `unsigned long`
- 2 Malé alokace (sta bajtů, max. až cca. 4M)
  - `kmalloc`, `kfree`

Alokace mají (většinou) GFP parametr. Ten určuje, co si alokátor může dovolit (spát, swapovat, použít `HIGHMEM`, ...). Prozatím nám stačí `GFP_KERNEL`.

```
void *mem1 = (void *)__get_free_page(GFP_KERNEL);
void *mem2 = kmalloc(PAGE_SIZE, GFP_KERNEL);
if (mem1 && mem2) {
    ...
}
kfree(mem2);
free_page(mem1);
```

- 1 V `exit` funkci naalokujte stránku
- 2 Udělejte do ní `strcpy` „Bye”
- 3 Vypište paměť jako řetězec
- 4 Uvolněte stránku
- 5 Vložte a odeberte modul ze systému
- 6 Jako obvykle `commit+push`