

# Sítové operační systémy

## Přehled předmětu

Dan Komosný

# Přehled

## 1 Účel

## 2 Informace

- Povinnosti a požadavky
- Zdroje informací
- Cvičení a projekt

## 3 Časté dotazy

# Přehled

1 Účel

2 Informace

3 Časté dotazy

# Cíl předmětu

Cílem je poskytnout **obecné** základy OS

- Teoretická část je zaměřena na **základní principy**
- Cvičení tyto principy **demonstrují**
- Projekt tyto principy **aplikuje**

# Cíl předmětu

Rozumět operačním systémům a věcem s tím spojených – programování, bezpečnost, reverzní inženýrství není o tom, že víte na co „kliknout“

- „Klikací“ přístup a znalost konkrétních aplikací je krátkozraká a na jedno použití (aplikace jsou – nebudou; budou jiné – brzy)
- Znalost principů má dlouhodobou hodnotu a má univerzální využití
- Praktická část je zaměřena na obecné nástroje, které lze kombinovat pro různé účely

# Obsah předmětu

Předmět je vyučován česky i anglicky (obecné označení OS)

- ① Sítové operační systémy – SOS
- ② Network operating systems – NOS

Lze používat i anglickou verzi

# Přednášky

- **Úvod** – vstup-výstup, strojový kód, assembler
- **Architektura** – typy, jádro, systémová volání, komplikace jádra, virtualizace
- **Procesy** – halda, zásobník, vlákna, stavy činnosti, plánování, komunikace mezi procesy, souběh, uvíznutí
- **Paměť** – statická paměť, virtuální paměť, stránkování, segmentace
- **Souborové systémy** – ukládání dat, metadata, žurnálovací systémy, virtuální soubory, organizace souborů
- **Síťový subsystém** – síťové rozhraní, servery, služby, bezpečnost

# Přednášky

## Přednášky – síťová část

- Uveden pouze základ síťové komunikace nutný pro popis **síťové části OS**
- Komplexní síťová komunikace je probírána v jiných předmětech

# Cvičení

- **Seznámení A** – příkazový řádek, skripty, pro cvičení a projekt
- **Seznámení B** – přístupová práva, správa uživatelů, pro cvičení a projekt
- **Modul jádra** – jádro, vytvoření modulu jádra
- **Virtualizace** – kontejnery, útok na hesla
- **Procesy** – správa procesů, automatická záloha
- **Konfigurace služeb** – konfigurační soubory OS, virtuální webový server
- **Start po síti** – zavaděč OS, spuštění hry po síti
- **Síťový substitém** – firewall/proxy, síťový útok

# Přehled

1 Účel

2 Informace

3 Časté dotazy

# Přehled

## 1 Účel

## 2 Informace

- Povinnosti a požadavky
- Zdroje informací
- Cvičení a projekt

## 3 Časté dotazy

# Povinnosti a požadavky

Co je povinné a co ne

- Přednášky jsou **nepovinné**
- Počítačová cvičení jsou **povinná**
- Projekt je **nepovinný**

# Povinnosti a požadavky

## Požadavky

- Předmět **nenavazuje** na žádný předchozí
- Požadavkem jsou základy programování
- Cvičení na počítači jsou v pořadí od **jednodušších** ke **složitějším**

# Přehled

## 1 Účel

## 2 Informace

- Povinnosti a požadavky
- **Zdroje informací**
- Cvičení a projekt

## 3 Časté dotazy

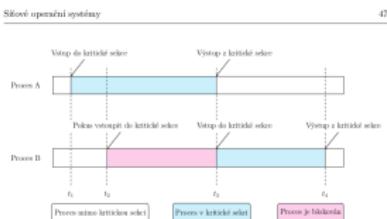
# Zdroje informací

## Zdroje informací

- Skripta – Sítové operační systémy
- Skripta – Počítačová cvičení předmětu Sítové operační systémy
- Projekt – samostatné zadání
- Další materiály – e-learning

# Skripta k teorii

## Skripta – k dispozici varianty pro snazší práci (tisk, čtení, poznámky)



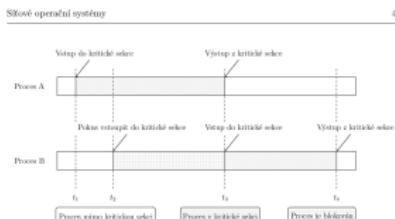
Obrázek 4.15: Kritické sekce procesů.

Pro zajistění vajíčkového vyloučení je do zdrojového kódu zařazena vstupní a výstupní sekce, viz výpis 4.8:

```
def
    kod procesu
        vstupni sekce
        ... 
        vystupni sekce
    kod procesu
    jumble(TIME);
```

Výpis kód 4.8: Vstupní a výstupní sekce pro zajistění vajíčkového vyloučení.

Obranou myšlenka vajíčkového vyloučení je postavena na principu zámku nad věžním prostředím. Proces před vstupem do kritické sekce zámek zamyká, a pokud je „otevřen“, nemůže „zamknout“ a vstoupit. Po vstupu z kritické sekce zámek „uvolní“ a další proces může vstoupit do své kritické sekce. Při popisu mnoha řešení se nejdříve zaměříme na intuitivní přístup a vysvětlíme si, v čem spočívá jejich problematické použití. Dále se dostaneme k řešením, která se používají.



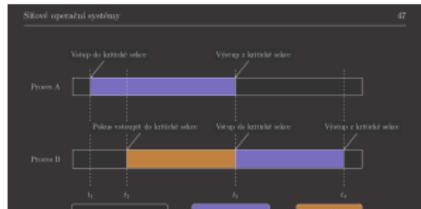
Obrázek 4.15: Kritické sekce procesů.

Pro zajistění vajíčkového vyloučení je do zdrojového kódu zařazena vstupní výstupní sekce, viz výpis 4.8:

```
def
    kod procesu
        vstupni sekce
        ...
        vystupni sekce
    kod procesu
    jumble(TIME);
```

Výpis kód 4.8: Vstupní a výstupní sekce pro zajistění vajíčkového vyloučení.

Obranou myšlenka vajíčkového vyloučení je postavena na principu zámku nad věžním prostředím. Proces před vstupem do kritické sekce zámek zamyká, a pokud je „otevřen“, nemůže „zamknout“ a vstoupit. Po vstupu z kritické sekce zámek „uvolní“ a další proces může vstoupit do své kritické sekce. Při popisu mnoha řešení se nejdříve zaměříme na intuitivní přístup a vysvětlíme si, v čem spočívá jejich problematické použití. Dále se dostaneme k řešením, která se používají.



Obrázek 4.15: Kritické sekce procesů.

Pro zajistění vajíčkového vyloučení je do zdrojového kódu zařazena vstupní výstupní sekce, viz výpis 4.8:

```
def
    kod procesu
        vstupni sekce
        ...
        vystupni sekce
    kod procesu
    jumble(TIME);
```

Výpis kód 4.8: Vstupní a výstupní sekce pro zajistění vajíčkového vyloučení.

Obranou myšlenka vajíčkového vyloučení je postavena na principu zámku nad věžním prostředím. Proces před vstupem do kritické sekce zámek zamyká, a pokud je „otevřen“, nemůže „zamknout“ a vstoupit. Po vstupu z kritické sekce zámek „uvolní“ a další proces může vstoupit do své kritické sekce. Při popisu mnoha řešení se nejdříve zaměříme na intuitivní přístup a vysvětlíme si, v čem spočívá jejich problematické použití. Dále se dostaneme k řešením, která se používají.

# Skripta k teorii

## Příklady

```
1  []$ nmap localhost
2  Nmap scan report for localhost (127.0.0.1)
3  Host is up (0.0000040s latency).
4  rDNS record for 127.0.0.1: localhost.localdomain
5  Not shown: 995 closed ports
6  PORT      STATE SERVICE
7  22/tcp    open  ssh
8  25/tcp    open  smtp
9  5900/tcp  open  vnc
10
11 Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

Výpis kódu 8.2: Přehled otevřených portů a svázaných služeb operačního systému.

# Skripta k počítačovým cvičením

Skripta cvičení – tři části: i) úvod, ii) vypracování, iii) samostatný úkol

## 5.3.2 Kompilace, vložení modulu do jádra, odstranění

### Vytvoření souboru **Makefile**

- Nyní bude provedena komplikace modulu. Pomocí editoru *vi* vytvořte soubor **Makefile** (velké písmeno M, jedno slovo). Do souboru vložíme následující text:

```
1 obj-m := hello.o
2 KDIR := /lib/modules/$(shell uname -r)/build
3 PWD := $(shell pwd)
4 default:
5     $(MAKE) -C $(KDIR) M=$(PWD) modules
6 clean:
7     $(MAKE) -C $(KDIR) M=$(PWD) clean
```

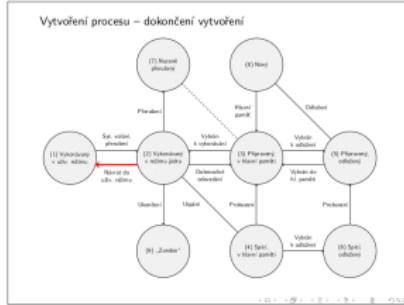
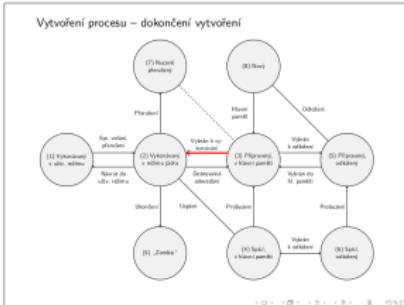
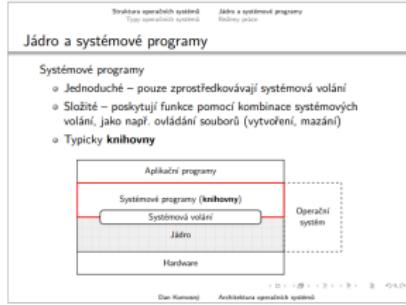
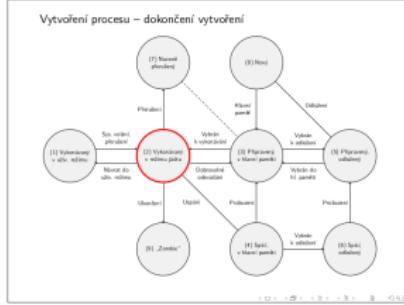
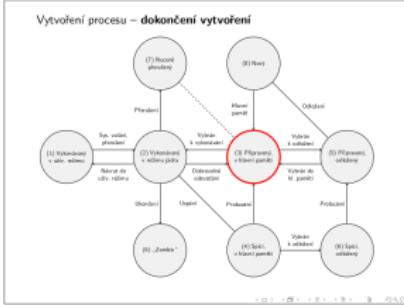
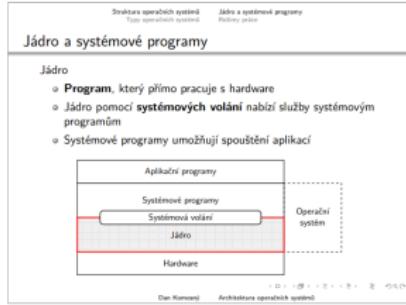
**Pozor:** Přípona souboru `hello` je písmeno „o“. Dále na místě `__` musí být znak TAB.

# Prezentace

## Prezentace

- Slouží pro výklad na přednášce – **nejsou primárním zdrojem** (jsou pomůcka pro přednášejícího)
- Jsou zhotoveny tak, aby **byly přehledné** – méně textu na slajdu, pomocné texty a obrázky pro lepší pochopení
- Obsahují grafická **rozšíření**, například průchody stavů procesů
- Počet sladů je **z toho důvodu větší** pro přehlednost a lepší porozumění

## Prezentace – poznámky a grafické prvky



# Přehled

## 1 Účel

## 2 Informace

- Povinnosti a požadavky
- Zdroje informací
- Cvičení a projekt

## 3 Časté dotazy

# Cvičení

## Pracovní prostředí pro cvičení

- Distribuce CentOS
- Cvičení jsou postavena tak, aby byla **nezávislá na distribuci**
- Praktické znalosti lze tak přímo nebo s menšími změnami využít v dalších OS

# Projekt



# Přehled

1 Účel

2 Informace

3 Časté dotazy

# Časté dotazy

Proč se v předmětu probírají **obecné základy** operačních systémů?

- Jedná se o první předmět oborů (kde je veden jako povinný), který se zabývá operačními systémy
- Získané znalosti lze rozšířit ve volitelných předmětech
- Je potřeba mít základy, pokud chcete něco vyřešit; bez základů „střílíte naslepo“

Proč se probírají systémová volání, podmínky souběhu, algoritmy práce paměti, žurnálovací souborové systémy, síťové sokety atd.?

- Jedná se o **základní stavební kameny** všech OS (jádro, procesy, paměť, ukládání dat, síťový substitém)

Kde můžu využít znalosti v praxi? – příklady

- Telekomunikace – síťové prvky a multimedialní zařízení, bezpečnost
- Robotika a senzorika – řídící SW, dedikované aplikace, minimalistické systémy, bezpečnost
- Letecký průmysl – systémy reálného času; systémy bez možnosti uvíznutí
- Informatika – výpočetní klastry, virtualizace, bezpečnost

# Časté dotazy – využití

Internet věcí

**Contiki**

The Open Source OS for the Internet of Things

TinyOS

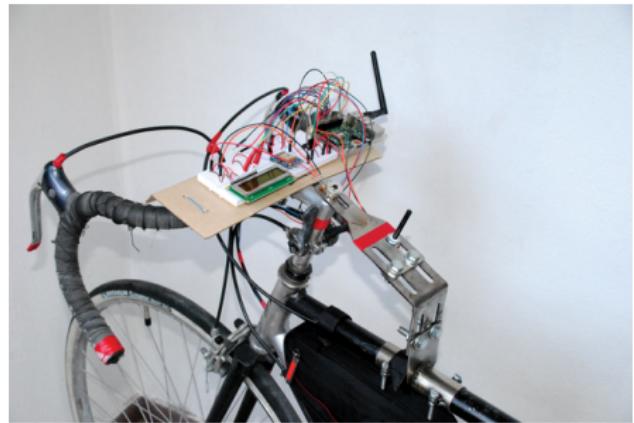


# Časté dotazy – využití

## Minimalistické systémy



[www.raspbian.org](http://www.raspbian.org)



# Časté dotazy – využití

Systémy reálného času; systémy bez uvíznutí



# Časté dotazy

Některá počítačová cvičení ??? (na první pohled)

- Proč se čte email v příkazové řádku? → Jedna ze základních věcí jak číst systémové zprávy
- Proč pracujeme s moduly? → Jádro pomocí modulů **podporuje různá HW rozhraní**

Předmět se částečně kryje s předmětem Bezpečnost ICT 1

- Předmět Bezpečnost ICT 1 vznikl nedávno a opakuje partie operačních systémů
- Na úpravě předmětu ICT 1 se pracuje

Konec prezentace