

# Základy operačních systémů

KIV/ZOS

# Kontaktní informace

- Ing. Ladislav Pešička
- UL401 (pozor, ne UK410)
- *pesicka@kiv.zcu.cz*
- Úřední hodiny
  - Út 9:30 až 10:30
  - St 9:30 až 10:30

# Web

- Stránky předmětu budou v portálu <http://moje.zcu.cz>
- V současné době je k dispozici <http://www.kiv.zcu.cz/~pesicka/zos> bude aktualizováno

# Požadavky na zápočet

- 2 zápočtové testy

- Každý 5.5 bodů z 10
- 1 náhradní

- 1. test

- Základy Linuxu, teorie z přednášek
- Cca 5.týden (bude na webu)

# Zkouška

## ■ Písemka

- Test na 60 min. bez pomůcek
- 11 bodů z 21 možných

## ■ Návrh známky

- Ústní konzultace

# OS - příklady

## ■ Servery, pracovní stanice

– Windows, Linux, Solaris

## ■ PDA, mobilní zařízení

– Windows CE, Symbian, Linux

## ■ Routery, AP, soho síťová zařízení

– Cisco IOS, Linux, VxWorks

## ■ Embedded zařízení

– Bankomaty, stravovací systémy, lékařské přístroje

– Windows CE, Windows XP embedded

# ZOS

- Obecné principy
  - Není zaměřen na 1 systém, vychází z Unixu
- KIV/OS
  - Pokračováním, Unix / Linux
- Není hodnocením, který OS je lepší
- Praxe
  - Základy práce s Linuxem
  - Práce se sdílenými zdroji, ošetření kritické sekce

# ZOS přednášky

- Úvod. Historie OS, rozdělení OS, architektura a komponenty OS
- Proces jako abstrakce. Implementace procesu. Konstrukce pro vytváření procesů.
- Problém kritické sekce. Spin-lock. Semaforey. Kooperace procesů.
- Implementace semaforů. Monitory a jejich implementace.
- Zprávy, RPC. Klasické problémy meziprocesové komunikace: Problém večeřících filosofů.
- Plánování procesů
- Plánování procesů v interaktivních systémech. Uváznutí (deadlock).
- Uváznutí: detekce a zotavení, dynamické zabránění, prevence. Správa hlavní paměti, základní mechanismy.
- Správa hlavní paměti -- virtuální paměť.
- Vstupy a výstupy.
- Systémy souborů.
- Bezpečnost v OS.
- Případová studie: UNIX (Linux).



# Co všechno tvoří OS?

- **Není všeobecná definice**
- **Vše co dodavatel poskytuje jako OS**
  - Windows ...
- **Program, běžící po celou dobu běhu výp. Systému**
  - Linux, moduly, zavádění na žádost
- **Linux - 2 000 000 řádků**
- **Win2000 - 30 000 000 řádků**

# OS

- PC vybaveny SW vrstvou nazývanou **operační systém**, jehož úlohou je spravovat HW a poskytovat k němu programům jednodušší rozhraní
- OS **zprostředkovává** aplikacím **přístup** k HW
- Většinou běží v tzv. **privilegovaném** režimu
  - Neplatí pro MS DOS, různé embedded systémy
  - Někdy část OS v uživatelském režimu
  - Interpretované systémy (JVM)
- Aplikace – v **uživatelském** režimu
- OS může zasahovat do běhu aplikací
- Aplikace může požádat OS o službu

# OS

- OS **koordinuje** a **poskytuje** služby aplikacím
  - Analogie – dopravní systém, vláda, ..
- OS je program, který slouží jako prostředník mezi aplikacemi a hw počítače.
- Dva základní pohledy na OS
- **Rozšířený stroj** (shora dolů)
- **Správce zdrojů** (zdola nahoru)

# OS jako rozšířený stroj

## ■ Holý počítač

- Primitivní a obtížně programovatelný (I/O)
- Např. disky ...
  - Práce s hlavičkou disku
  - Alokace dealokace bloků dat
  - Více programů chce sdílet stejné médium

## ■ Chceme

- Jednoduchý pohled – pojmenované soubory
- OS skrývá před aplikacemi pravdu o HW (přerušování, správu paměti..)

# OS jako rozšířený stroj

- Strojové instrukce (holý stroj)
- **Vysokourovňové služby** (rozšířené instrukce)
  - Systémová volání
- Z pohledu programátora
  - Pojmenované soubory
  - Neomezená paměť
  - Transparentní I/O operace
- ZOS zkoumá, jako jsou služby v OS implementovány

# OS jako správce zdrojů

- OS jako poskytovatel / správce zdrojů (resource manager)
- Různé zdroje (čas CPU, paměť, I/O zařízení)
- OS – správná a řízená **alokace** zdrojů programům, které je požadují (přístupová práva)
- **Konfliktní požadavky** na zdroje
  - V jakém pořadí vyřízeny
  - Efektivnost, spravedlivost

# Historický vývoj

- Vývoj hw -> vývoj OS
- 1. počítač – ENIAC, 15.2.1946
  - Tělocvična
  - 18 000 elektronek
  - Regály, chlazení – letecké motory
  - 5000 operací/s

# Generace počítačů

1. Elektronky
2. Tranzistory
3. Integrované obvody
4. LSI, VLSI (mikroprocesory,..)



# 1. Generace (1945-55)

- Elektronky, propojovací desky
- Programování
  - V absolutním jazyce
  - Propojování zdířek na desce
  - Později děrné štítky, assemblery, knihovny, FORTRAN
  - Numerické kalkulace
- Způsob práce
  - Stejní lidé – stroj navrhli, postavili, programovali
  - Zatrhnout blok času na rozvrhu, doufat, že to vyjde
- OS ještě neexistují

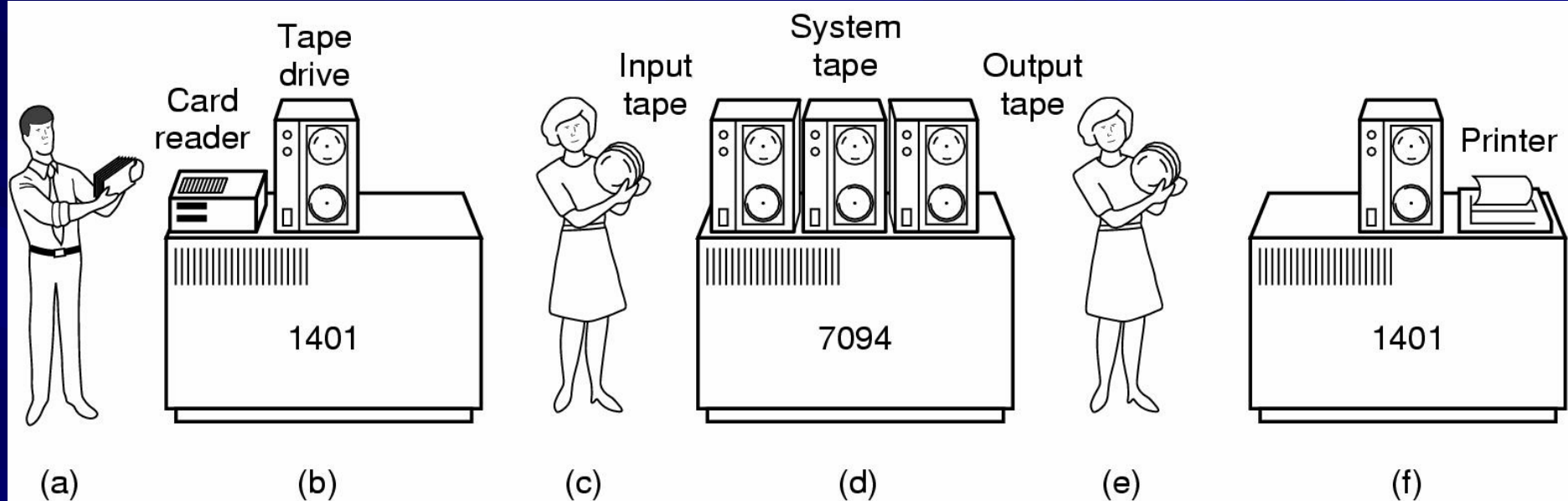
## 2. Generace (1955-65)

- Tranzistory, dávkové OS
- Vyšší spolehlivost; klimatizované sály
- Oddělení návrhářů, výroby, operátorů, programátorů, údržby
- Mil \$ - velké firmy, vlády, univerzity
- Způsob práce
  - Vyděrovat štítky s programem
  - Krabici dát operátorovi
  - Výsledek vytisknut na tiskárně
- Optimalizace
  - Na levném stroji štítky přenést na magnetickou pásku

## 2. generace

- Sekvenční vykonávání dávek
- Ochrana systému – kdokoliv dokázal shodit
- OS IBSYS = IBM SYSTÉM FOR 7094
- Pokud úloha prováděla I/O, CPU čekal..
- Viz Tanenbaum

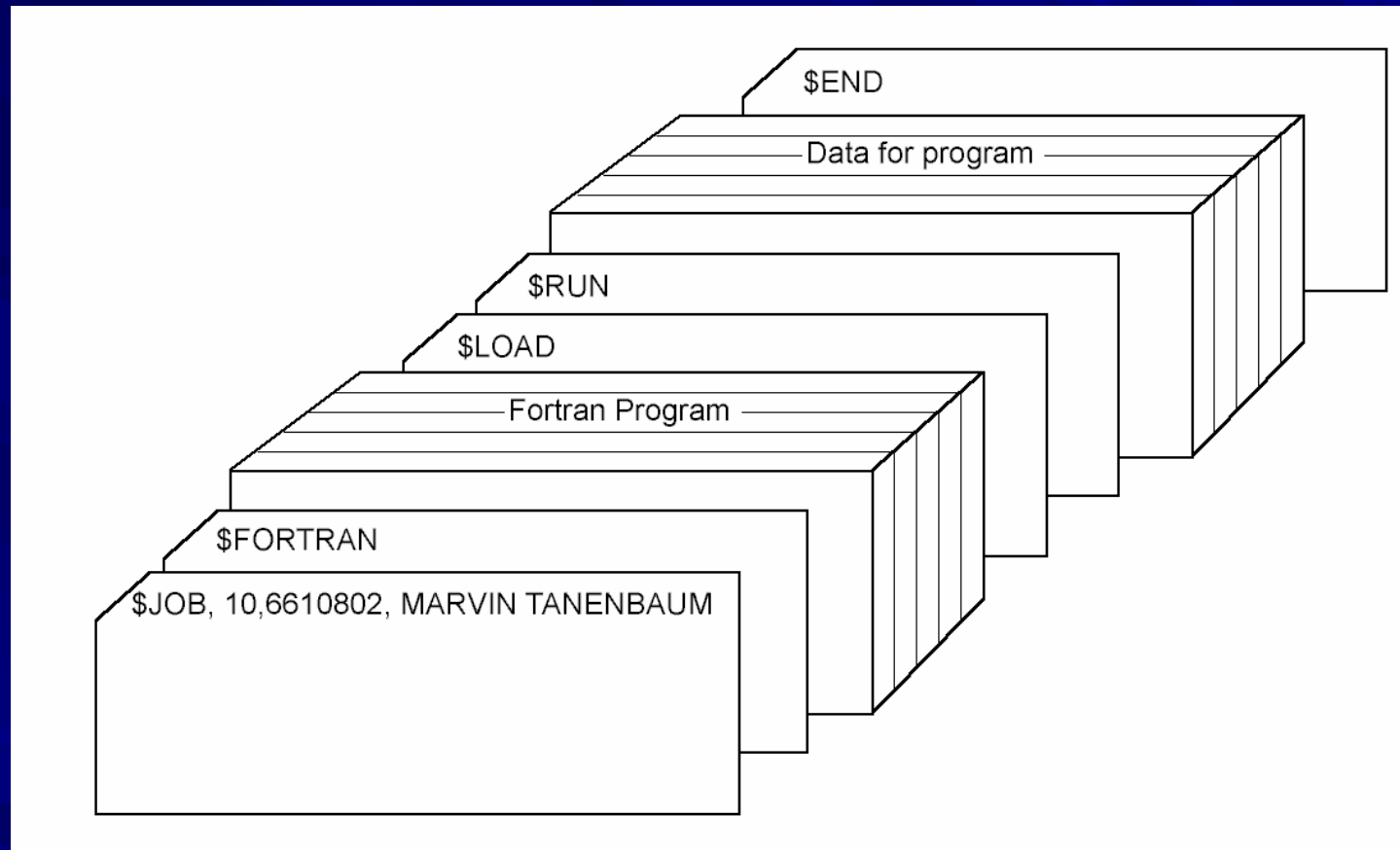
# History of Operating Systems (1)



## Early batch system

- bring cards to 1401
- read cards to tape
- put tape on 7094 which does computing
- put tape on 1401 which prints output

# History of Operating Systems (3)



- Structure of a typical FMS job – 2<sup>nd</sup> generation

# 3. Generace (1965-80)

- Integrované obvody, multiprogramování
- 2 řady počítačů
  - Vědecké výpočty
  - Komerční stroje – banky, pojišťovny
- IBM 360 – sjednocení
  - Malé i velké stroje
  - Komplexnost – spousta chyb

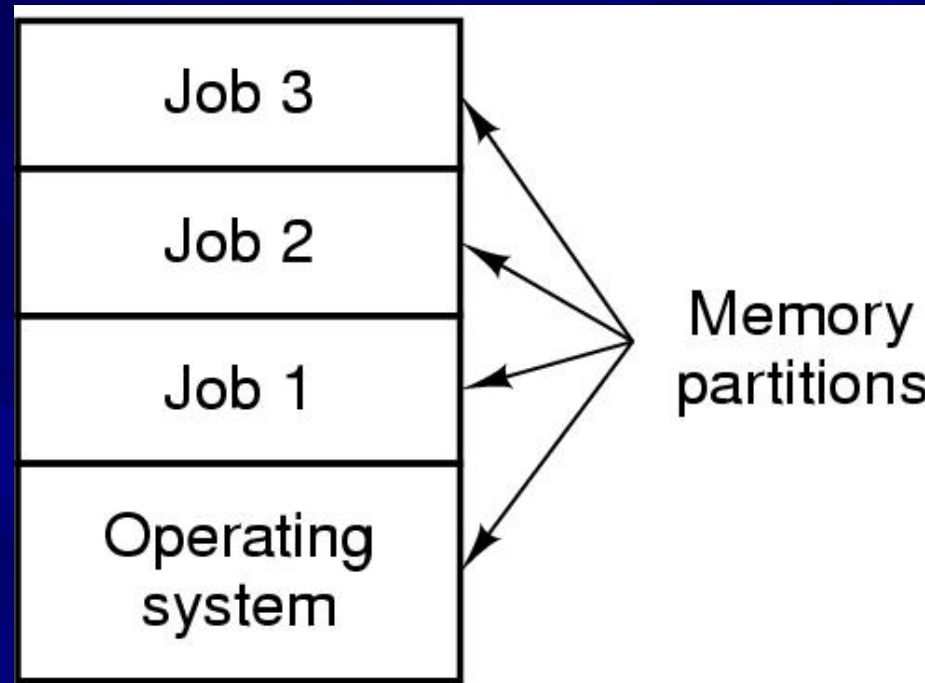
# 3. generace

## ■ Multiprogramování

- Doba čekání na I/O neefektivní  
(věda OK, banky 80-90% čekání)
- Více úloh v paměti
  - Napřed konstantní počet
  - HW pro ochranu paměti

## ■ Každá úloha ve vlastní oblasti paměti; zatímco jedna I/O, druhá počítá ...

# History of Operating Systems (4)



- Multiprogramming system
  - three jobs in memory – 3<sup>rd</sup> generation



# 3. generace

## ■ Spooling

- Na vstupu – ze štítků na disk, úloha se zavede z disku
- Na výstupu – výsledky na disk před výtiskem na tiskárně

## ■ Stále *dávkové systémy*

- Dodání úlohy, výsledek – několik hodin

# 3. generace

- **Systemy se sdílením času**  
(time shared system)
  - Varianta multiprogramování
  - CPU střídavě vykonává úlohy
  - Každý uživatel má on-line terminál
- CTSS (MIT 1962)
- MULTICS

# Minipočítače

- DEC PDP (1961)
  - Cca 3.5 mil Kč , „jako housky“
  - Až PDP11 – nekompatibilní navzájem
- Výzkumník Bell Labs pracující na MULTICSu  
**Ken Thompson** – našel nepoužívanou PDP-7,  
napsal omezenou jednouživat. Verzi MULTICSu  
vznik UNIXu a jazyka C (1969)

# 4. Generace (1980)

- Mikroprocesory, PC
- GUI x CLI
- Síťové a distribuované systémy
- MS DOS, Unix, Windows NT
- UNIX – dominantní na nonIntel;
- Linux, BSD – rozšíření i na PC
  - Výzkum Xerox PARC – vznik GUI
  - Apple Macintosh

# Dělení OS

- Dle úrovně sdílení CPU

- Jednoprocesový

- MS DOS, v daném čase v paměti aktivní 1 program

- Multiprocesový

- Efektivnost využití zdrojů
- Práce více uživatelů

# Dělení OS

## ■ Dle typu interakce

### ■ Dávkový systém

- Sekvenční dávky, není interakce

### ■ Interaktivní

- Interakce uživatel – úloha
- Víceprocesové – interakce max. do několika sekund (Win, Linux, ..)

# OS reálného času

- Výsledek má smysl, pouze pokud je získán v nějakém čase
- Aplikace přísné požadavky na čas odpovědi
  - Řídící počítače, multimedia
- Časově ohraničené požadavky na odpověď
  - Řízení válcovny plechu, výtahu ☺
- Nejlepší snaha systému
  - Multimedia, virtuální realita

# Hard realtime OS

- Zaručena odezva v ohraničeném čase
- Včetně zpoždění a režie systému ohraničeny
- Omezení OS
- Často není systém souborů, virtuální paměť
- Nelze zároveň sdílení času
- Řízení výroby, robotika, telekomunikace



# Soft RT system

- Priorita RT úloh před ostatními
- Nezaručuje odezvu v daném čase
- Lze v systémech sdílení času
- RT Linux
- Multimédia, virtuální realita

# Další dělení OS

- Dle velikosti HW
  - Superpočítač, telefon, čipová karta
- Míra distribuovanosti
  - Klasické - centralizované 1 and more CPU
  - Paralelní
  - Síťové
  - Distribuované
    - virtuální uniprocessor
    - Uživatel neví kde běží programy, kde jsou soubory

# Další dělení OS

- Podle počtu uživatelů
  - Jedno a víceuživatelské
  
- Podle funkcí
  - Univerzální
  - Specializované (IOS)

# Pokračování

- dále viz pdf soubor k přednášce č.1  
(*p1zaklad.pdf*)

# Literatura, použité zdroje

Obrázky z některých slidů (20, 21, 24) pocházejí z knížky

Andrew S. Tanenbaum: Modern Operating Systems

vřele doporučuji tuto knihu, nebo se alespoň podívat na slidy ke knize dostupné mj. na webu předmětu v *Přednášky -> Odkazy*