

{ Pr-6 6 }

19. Operační systémy

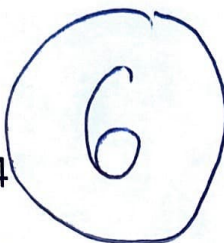
Programování - 6
Verze 1.2

David Martinek, 2016-2024

Gymnázium Brno, Vídeňská



Prezentace pro výuku programování, jejímž autorem je Ing. David Martinek, podléhá licenci Creative Commons Uveďte autora-Neužívejte dílo komerčně-Zachovejte licenci 4.0 Mezinárodní.



{ Pr-6 6 }

Operační systémy

- Stručná historie OS
- Operační systém
- Základní části OS
- Základní úlohy OS
- Druhy OS
- OS jako správce prostředků
- Modulární OS
- Základní moduly OS
- Úrovně OS
- Proces - základní pojmy
- Proces
- Vlákno
- Multitasking
- Životní cyklus procesu

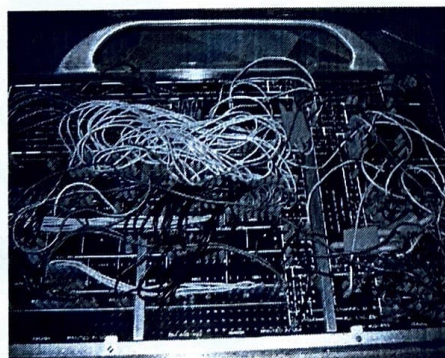
Programování - teorie, 6. ročník

2/48

{ Pr-6 6 }

Stručná historie OS

- 40. léta 20. století
 - Jednoúlohové počítače
 - Programují se pomocí propojovací desky (plugboard).
 - Zapojení jedné úlohy trvá v řádu týdnů!
 - Bez operačního systému



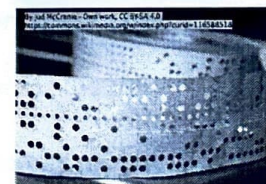
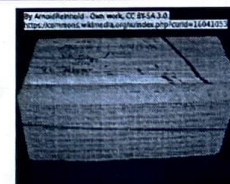
CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=522789>

3/48

{ Pr-6 6 }

Stručná historie OS

- 50.-60. léta 20. století
 - Přepínací panely
 - Vylepšení propojovacích desek
 - Děrné štítky, pásky
 - Nejprve pro vstup i výstup dat, pak i programů



Programování - teorie, 6. ročník

4/48

{ Pr-6 6 }

Stručná historie OS

- 50.-60. léta 20. století
 - Počítačový terminál
 - Využití technologie dálnopisu (teleprinter, telex) → komunikace přes elektrický psací stroj
 - Později televizní obrazovka s textovým rozhraním (koncem 60. let)



5/48

Programování - teorie, 6. ročník

{ Pr-6 6 }

Stručná historie OS

- 50.-60. léta 20. století
 - Sálkové počítače (mainframe)
 - Dávkové zpracování
 - Úlohy seřazeny za sebou
 - Jakmile jedna skončí, začne se zpracovávat další.
 - První náznaky operačních systémů
 - Chytrým přeuspořádáním částí programů v dávkě lze efektivněji využívat počítačový čas.



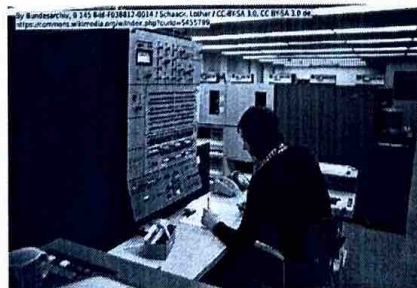
6/48

Programování - teorie, 6. ročník

{ Pr-6 6 }

Stručná historie OS

- 1964: IBM OS/360
 - Dávkový operační systém pro počítač IBM System/360
 - První verze pracovaly ještě s děrnými štítky.
 - V 60. letech velmi rozšířený
 - Používan např. u NASA pro program Apollo, nebo firmami jako Volkswagen.



7/48

Programování - teorie, 6. ročník

{ Pr-6 6 }

Stručná historie OS

- 1964: Zahájen vývoj Multicsu
 - Společný projekt MIT, General Electric a Bell Labs
 - Předchůdce dnešních OS
 - Nebyl vázán na jediný počítač.
 - Modulární HW a SW - výkon lze zvýšit přidáváním komponent i za plného provozu (!)
 - Segmentování paměti - předchůdce segmentového adresování
 - Zavedl koncept multitaskingu.
 - Přímou inspiroval vznik Unixu a tím i dalších moderních OS.

Programování - teorie, 6. ročník

8/48

{ Pr-6 6 }

Stručná historie OS

- 1969: Unix
 - Ken Thompson a Dennis Ritchie, Bell labs, AT&T
 - Napsán v jazyce C (společně s ním).
 - Obecný, víceúlohový (s preemptivním multitaskingem), víceuživatelský, síťový, ...
 - Nejvlivnější SW projekt 20. století
 - Vyvíjen a používán dodnes



Ken Thompson



Dennis Ritchie

9/48

Programování - teorie, 6. ročník

{ Pr-6 6 }

Stručná historie OS

- 1978: BSD Unix
 - Univerzitní větev Unixu
- 1981: MS DOS \Rightarrow SRAČKA
 - Extrémně zjednodušený, Unixem inspirovaný OS pro IBM PC
 - První IBM PC byly velice slabé počítače v porovnání se sálovými počítači s Unixem.
- 1984: Mac OS
 - Classic Mac OS běžící na procesorech PowerPC
- 1985: MS Windows v. 1.0

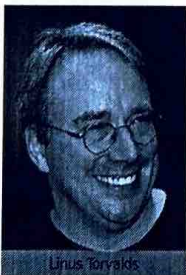
Programování - teorie, 6. ročník

10/48

{ Pr-6 6 }

Stručná historie OS

- 1991: Linux
 - Reinterpretace (jádra) Unixu od Linuse Torvaldse pro PC napsaná v C.
 - Původně šlo o hobby/školní projekt.
 - PC byly v té době považovány za příliš primitivní pro provozování Unixu. Unix byl doma na sálových počítačích.
 - Projekt GNU (1984) + Linux = GNU/Linux
- 1995: MS Windows 95
- 2001: Mac OS X
 - Jádro založeno na BSD Unixu



Linus Torvalds

11/48

Programování - teorie, 6. ročník

{ Pr-6 6 }

Stručná historie OS

- 2007: iOS
 - OS firmy Apple pro mobilní zařízení (iPhone, iPod, iPad)
 - BSD jádro (Darwin) + GUI
 - C, C++, Objective-C, Swift
- 2008: Android
 - OS firmy Google pro mobilní zařízení (mobily, tablety různých výrobců)
 - Linuxové jádro + Java virtual machine
 - Java, C, C++

Programování - teorie, 6. ročník

12/48

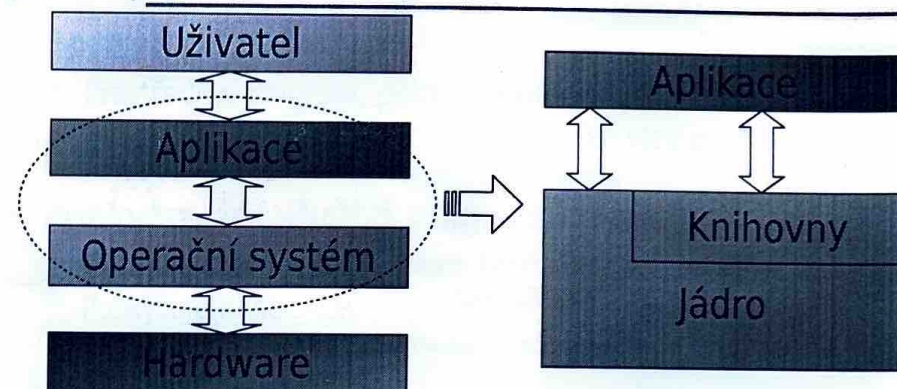
{ Pr-6 6 }

Operační systém

- Základní programové vybavení počítače
- OS je program (sada programů).
 - Mezivrstva mezi HW a uživatelskými programy a uživateli
 - Umožňuje uživateli pohodlně a efektivně používat počítač.
 - Umožňuje a řídí běh ostatních programů.

{ Pr-6 6 }

Operační systém



{ Pr-6 6 }

Operační systém

- Zavádí se do paměti hned po startu počítače.
 - Nejdříve jádro, až pak CLI a/nebo GUI
- Běží až do vypnutí počítače.
- Obvyklé úlohy
 - Správa procesů, přerušení, přidělování procesoru, přidělování a ochrana paměti, souborový systém, ovládání zařízení, síť, bezpečnost/řízení přístupu, vstup/výstup

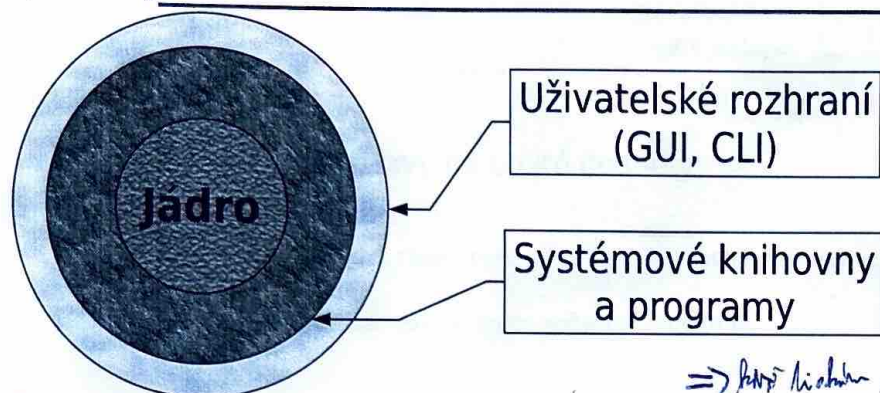
{ Pr-6 6 }

Základní části OS

- **Jádرو** (kernel)
 - Moduly pro přidělování prostředků, řízení procesů, ...
- **Systémové knihovny a programy**
- **Uživatelské rozhraní**
 - CLI - textové rozhraní (ls, dir, mkdir, ...)
 - GUI - grafické rozhraní (správce oken, správce souborů, dialogy, ...)
 - Operační systém není jen GUI!

{ Pr-6
6 }

*jádro => jeden z největších problémů na osin PC
-> musí být osin modulární*
Základní části OS



Programování - teorie, 6. ročník

17/48

{ Pr-6
6 }

Základní úlohy OS

• Správa prostředků

- Prostředky: procesor, paměť, periferie (*grafická karta, ...*)
- Efektivní a bezpečné sdílení prostředků více programy, uživateli

• Poskytování služeb programům a uživatelům

- Definuje standardní rozhraní (knihovny, uživatelské rozhraní).
- Poskytuje abstrakce. *-> jak o něm vidím procesor*
 - Zjednodušuje pohled na hardware a činnosti počítače.
 - Příklady: soubor, proces

Programování - teorie, 6. ročník

18/48

{ Pr-6
6 }

Druhy OS ! ZAPAMATOVAT !

• Jednoúlohový

- V jednom čase běží jen jedna úloha. *MS DOS*
- Dávkový systém
 - Až skončí jedna úloha, poběží další naplánovaná úloha. (*úloha a kromě toho...*)

• Víceúlohový (*lepší pro lidi*)

- V jednom čase běží více úloh současně.
- **Multitasking - současný běh více úloh, než je procesorů.**
 - Přepíná běh několika úloh tak rychle, že se zdá, jako by běžely paralelně.
 - Kooperativní vs. preemptivní

Programování - teorie, 6. ročník

19/48

{ Pr-6
6 }

Druhy OS

• Jednouživatelský

- dříve MS DOS, herní konzole (Atari, ZX Spectrum)
- dnes mobily (tzv. „bez OS“, starší verze „smart“: Symbian, Android, iOS)

• Víceuživatelský

- Uživatelé mají bezpečně oddělené účty - proces jednoho uživatele nemůže ohrozit procesy ani data jiného uživatele
- Unix, Linux, MS Windows, Mac OS X, ...

Programování - teorie, 6. ročník

20/48

{ Pr-6 6 }

Druhy OS

• Real-time

- Garantuje dobu odezvy na určité podněty.
- Pro speciální použití
 - Řídící systémy letadel, raket, ponorek, některých továren, jaderných elektráren
 - Všude, kde by pozdní reakce mohly způsobit katastrofu.

jaderné elektrárny ⇒ musí tam být Windows (alternativně)

{ Pr-6 6 }

Modulární OS

- OS jsou velké SW projekty ⇒ bez modulů nezvládnutelné.
- Moduly řeší různé aspekty fungování OS.
 - Správa prostředků, ovládání HW zařízení
- Snaha o zefektivnění vývoje.

{ Pr-6 6 }

OS jako správce prostředků

- Pracuje v **privilegovaném režimu procesoru**.!
 - Může přerušovat a obnovovat běh ostatních programů.
 - Musí mít podporu v hardware - speciální instrukce a přerušení procesoru. *OS vyžaduje speciální procesor!*
- **Udržuje přehled** o všech sdílených prostředcích.
- **Plánuje a rozhoduje** komu, kdy a kolik prostředků přidělit.
- **Přiděluje a vyžaduje navrácení** prostředků.
 - U kritických prostředků *vynucuje* navrácení (např. procesorový čas).

(připravení procesor ⇒ vyžaduje přerušení)

{ Pr-6 6 }

Typy modulárních OS

- **Monolitické jádro**
 - Jádro je kompaktní program. ⇒ *přidělí se to do jedné velké binární souboru*
- **Mikrojádro**
 - Malé jádro + samostatné servery (moduly) ⇒ *musí být binární (binární dílčí soubory)* → *mechanizmus*
- **Hybridní jádro**
 - Kombinace předchozích přístupů

{ Pr-6
6 }

Monolitické jádro

- Moduly na úrovni zdrojového kódu
- Po sestavení vzniká kompaktní binární program - jádro.
- Příklady: Unix, Linux

{ Pr-6
6 }

Monolitické jádro

- Výhody
 - Vyšší efektivita
 - Moduly sdílejí datové struktury.
 - Moduly spolu rychleji komunikují.
 - Snazší vývoj (?)
 - Přání vývojářů...
- Nevýhody
 - Všechny moduly *musí* běžet v chráněném režimu
 - Bezpečnost - havárie jednoho modulu vede k havárii celého systému (blue screen, kernel panic)
 - Obtíže s přidáváním modulů za běhu systému.
 - Od určité velikosti obtížný vývoj.
 - Zatím se ale stále daří jej zvládat.

{ Pr-6
6 }

Mikrojádru

- Malé základní jádro
 - Běží v privilegovaném režimu.
- Moduly
 - Tzv. servery obsluhující nekritické služby systému.
- Příklady
 - Hurd, zatím jen akademické projekty

{ Pr-6
6 }

Mikrojádru

- Výhody
 - Dynamická změna modulů za běhu
 - Dnes to umí i monolity...
 - Teoreticky vyšší bezpečnost
 - Ale na úkor efektivity
 - Teoreticky snazší vývoj (???)
- Nevýhody
 - Menší výkon *⇒ normální*
 - Komunikace mezi moduly,
 - Oddělené datové struktury ⇒ nutnost duplikace, synchronizace
 - V praxi se zatím příliš tento přístup neuchytil.

{ Pr-6 6 }

Hybridní

- Oba předchozí přístupy mají své nevýhody.
- Hybridní – snaha vzít z obou přístupů to dobré a minimalizovat nevýhody
 - MS Windows (daří se to ???)
- Který z těchto tří přístupů do budoucna zvítězí není zcela jasné.
 - Paralelní a distribuovaný HW generuje nové problémy.

Programování - teorie, 6. ročník

29/48

{ Pr-6 6 }

Modul přidělování procesoru

- Vyžaduje *privilegovaný režim procesoru*.
 - Pro preemptivní multitasking (viz dále)
- **Plánovač úloh** (část modulu přidělování procesoru)
 - rozhoduje o přepínání procesů,
 - rozhoduje o časových úsecích, které jsou procesu přiřazeny,
 - bere v úvahu priority procesů, jejich požadavky, aktuální stav.

Programování - teorie, 6. ročník

31/48

procesy o min. dobu
a přepínání se na fyzické

{ Pr-6 6 }

Základní moduly OS

- V nějaké podobě je najdeme u všech typů OS.

- Modul přidělování procesoru
- Modul přidělování paměti
- Modul správy souborů
- Modul správy periférií

některé možnosti (viz 1000)

Programování - teorie, 6. ročník

30/48

– pro stack ⇒ stack je 4kbit adresní interval min. 1000 a má se
– pro stack ⇒ stack je 4kbit adresní interval min. 1000 a má se

* když OS spouští, tak jde první min. 1000, pak je program

{ Pr-6 6 }

Modul přidělování paměti

- **Udržuje údaje o stavu paměti**
 - Celkový stav vs. paměť přidělená jednotlivým procesům
- **Přiděluje paměť** procesům, které o ni požádají
 - malloc, new
- **Uvolňuje paměť** procesů, které se jí vzdají
 - free, delete

Programování - teorie, 6. ročník

32/48



{ Pr-6 6 }

Modul přidělování paměti

- Přiděluje/uvolňuje paměť vznikajícím/zanikajícím procesům během zavádění do paměti a přerušení.
- Obsluhuje tzv. **chráněný režim**. *(chráněný režim se používá v počítačové paměti)* *
 - Přístup mimo přidělenou paměť způsobí HW **přerušení**,
 - které OS vyřeší zabitím provinilého procesu.

Programování - teorie, 6. ročník

mechanismus, kdy OS (i proces) spravuje přístup na kteroukoli paměťovou stránku

33/48

{ Pr-6 6 }

Modul správy souborů

- **Sleduje** stav souborů
- **Rozhoduje**
 - o přístupu procesů k souborům na základě jejich práv - ochrana souborů,
 - o umístění souborů na disku (paměťovém médiu)
- **Přiděluje** soubor pro použití (otevření)
- **Vyžaduje** vrácení souboru (zavření)

Programování - teorie, 6. ročník

34/48

{ Pr-6 6 }

Souborové systémy

- Moduly realizující souborový V/V
 - Pro různá fyzická zařízení (pevný disk, DVD, SSD, Flash, Ramdisk atd.)
 - V různých logických formátech (EXT4, FAT32, NTFS atd.)
- Logický formát dat
 - Soubory se dělí na bloky dat → nemusí být ukládány souvisle → různé datové struktury (tabulka, spojový seznam, stromové struktury).

Programování - teorie, 6. ročník

35/48

{ Pr-6 6 }

Modul správy periférií

- Rozhoduje o efektivním přidělování, plánuje V/V operace.
- Přiděluje, zahajuje V/V, vyžaduje vrácení.
- Realizuje virtualizaci zařízení
 - pomocí cache, vydává periférii za soubor atd.
- Proč?
 - Periférie jsou různě rychlé.
 - Jsou od různých výrobců - liší se v ovládání (každá má svůj ovladač).
- **Dispečer** *(program)*
 - Sleduje stav periférie (v HW smyslu).
- **Spooler** *(program)*
 - realizuje frontu požadavků (např. úloh pro tiskárnu)
- *bluž nemůže provést paralelně např. listování*

Programování - teorie, 6. ročník

36/48

{ Pr-6
6 }

Úrovně OS

- OS je hierarchický
 - Moduly tvoří jádro systému.
 - Moduly smí využívat služeb modulů na nižší úrovni a nesmí využívat moduly na vyšší úrovni.

{ Pr-6
6 }

Proces - základní pojmy

- Úloha
 - Souhrn činností potřebných k provedení určité zakázky.
- Krok úlohy
 - Jednotka činnosti
 - Provádějí se postupně, v určitém pořadí

{ Pr-6
6 }

Úrovně OS

- Úroveň 1 - modul přidělování procesoru na nižší úrovni
 - Plánovač procesů (žádosti o, přidělení, uvolnění prostředků)
- Úroveň 2 - modul přidělování paměti
- Úroveň 3 - modul přidělování procesoru na vyšší úrovni
 - Správa procesů, zasílání zpráv mezi procesy
- Úroveň 4 - modul přidělování periférií
- Úroveň 5 - modul správy souborů

komunikace mezi úrovněmi

- proces smí ⇒ dostávat dané služby

*vytváří je OS (proces používá OS o daném procesu ⇒ pomáhá)
které poskytuje*

{ Pr-6
6 }

Proces - základní pojmy

abstraktní model

- **Proces** (*živý program*) *→ živý ⇒ program běží ⇒ živý proces*
 - Výpočet, který lze provádět paralelně s ostatními výpočty.
 - Vytváří jej OS. *→ online regulérní přepínání*
- **Adresový prostor**
 - Oblast paměti, která je přidělena procesu.
 - Vymezuje legální adresy procesu (použije-li jiné, bude zabít).

proces si provádí pomocí OS (který přidělí mu OS)

→ online regulérní a regulérní

{ Pr-6
6 }

control Qh

Proces - základní pojmy

• Přerušeni

signály

(periodická kontrola ⇒ rozložení času ⇒ měření času nebo měření rychlosti)

- Mechanismus, díky němuž procesor přeruší svou činnost při určité události.
- Obsluha přerušeni
 - Předem registrovaná činnost (funkce), která je vyvolána v reakci na přerušeni.
 - Obvykle je má pod palcem OS.

{ Pr-6
6 }

Proces

- Spuštěný počítačový program
 - v operační paměti, využívá procesor („živý“ program).
- Vykonávaný kód + jeho (dynamicky se měnící) data.
- OS zajišťuje ochranu paměti
 - procesy si nemohou vzájemně škodit
- Je řízen a spravován OS (viz moduly)
- OS provozuje více procesů zároveň - přiděluje jim procesorový čas → multitasking.

{ Pr-6
6 }

Vlákno

• Odlehčený proces

- resp. proces se může dělit na vlákna.

! paralelní

• Vlákna sdílejí paměť procesu.

- Menší režie OS při paralelním běhu
- Nutnost řešit synchronizační problémy
 - Co když chce jedno vlákno proměnnou zapsat a zároveň jiné přečíst?
 - Deadlock, race condition
- Ne všechny OS vlákna podporují.

{ Pr-6
6 }

Multitasking

- Metoda přepínání procesů tak, aby se zdálo, že běží současně
 - i na jednom procesoru,
 - na více procesorech - současně běžících úloh je obvykle více než dostupných procesorů.
- Výměna vykonávaných procesů → **přepnutí kontextu**
 - Jádro vymění obsah bazových registrů a IP (instruction pointer) tak, aby ukazovaly na paměť a kód dalšího procesu
- Dvě základní strategie
 - Kooperativní vs. preemptivní

→ menší množství adresné paměti

Kooperativní multitasking

- Schéma chování
 - OS přiděluje procesu procesor.
 - Proces se po chvíli **sám vzdá procesoru** (zavolá funkci).
 - OS pak může přidělit procesor jinému procesu.
- Velmi nebezpečné
 - Co s chamtivými procesy? (viz MS Windows 3.11)
- Důvod vzniku
 - Primitivní HW – procesory bez podpory pro přerušování procesů

Programování - teorie, 6. ročník

45/48

$$\left\{ \begin{array}{c} \text{Pr-6} \\ 6 \end{array} \right\}$$

Preemptivní multitasking

- **Výhody**
 - Havárie jedného procesu neohrozí jiné procesy ani OS.
 - Vyšší bezpečnosť
 - Vyšší plynulosť
- **Nevýhody**
 - Vyžaduje podporu v HW.
 - Procesory určené pro běh OS mají privilegovaný režim.
 - Kontrolery takového režimu nemají implementován.
 - Složitější algoritmus v OS (jen drobný problém výrobce)

Programování - teorie, 6. ročník

$$\left\{ \begin{array}{c} \text{Pr-6} \\ 6 \end{array} \right\}$$

Preemptivní multitasking

- Schéma chování
 - **Jádro vynucuje přerušení** běhu procesů (řádově 100 - 1000× za sekundu),
 - nastaví v procesoru časovač, pak dojde k vynucenému *přerušení* běhu,
 - následně spouští další proces → přepnutí kontextu.
- Musí mít podporu v HW (procesoru)
 - Jádro běží v **privilegovaném režimu**.
 - Může používat speciální instrukce, může zasahovat do chráněných oblastí paměti procesů.
 - Procesy samotné tu část procesoru, která přerušuje běh procesů využívat nesmějí (procesor jim to nedovolí).

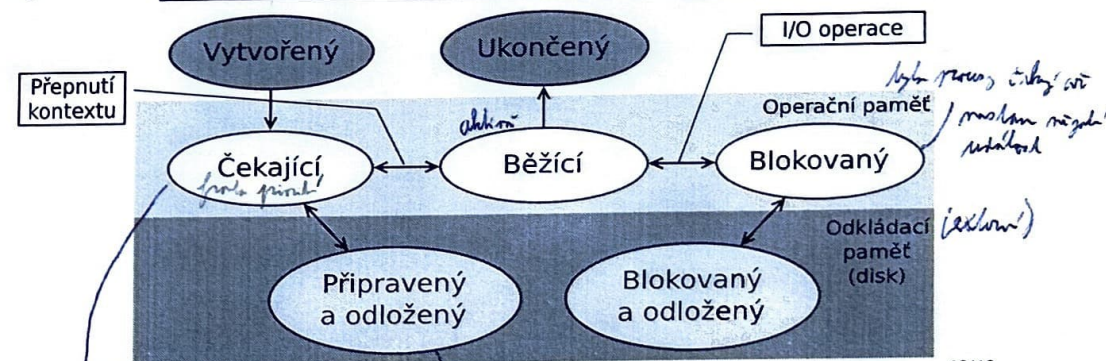
Programování - teorie, 6. ročník

46/48

pour le bien fait \Rightarrow r UNIT j'en le DÉMONI (m W, MAOS \Rightarrow rous d'air)
proug

$$\left\{ \begin{array}{c} \text{Pr-6} \\ 6 \end{array} \right\}$$

Životní cyklus procesu



Programování - teorie, 6. ročník

48/48

Erreichte \Rightarrow 'erreicht' abstrakt
 ding. konkret 'operational' haben

↳ belajar men. nilai positif, baik di pda kehidupan proses