

Procesy a vlákna

- Procesy
- Vznik a zánik procesu
- Životný cyklus procesu (stavový diagram)
- Vlákna (threads)
- Ukážky

Autor: Peter Tomcsányi

Niektoré práva vyhradené v zmysle licencie Creative Commons

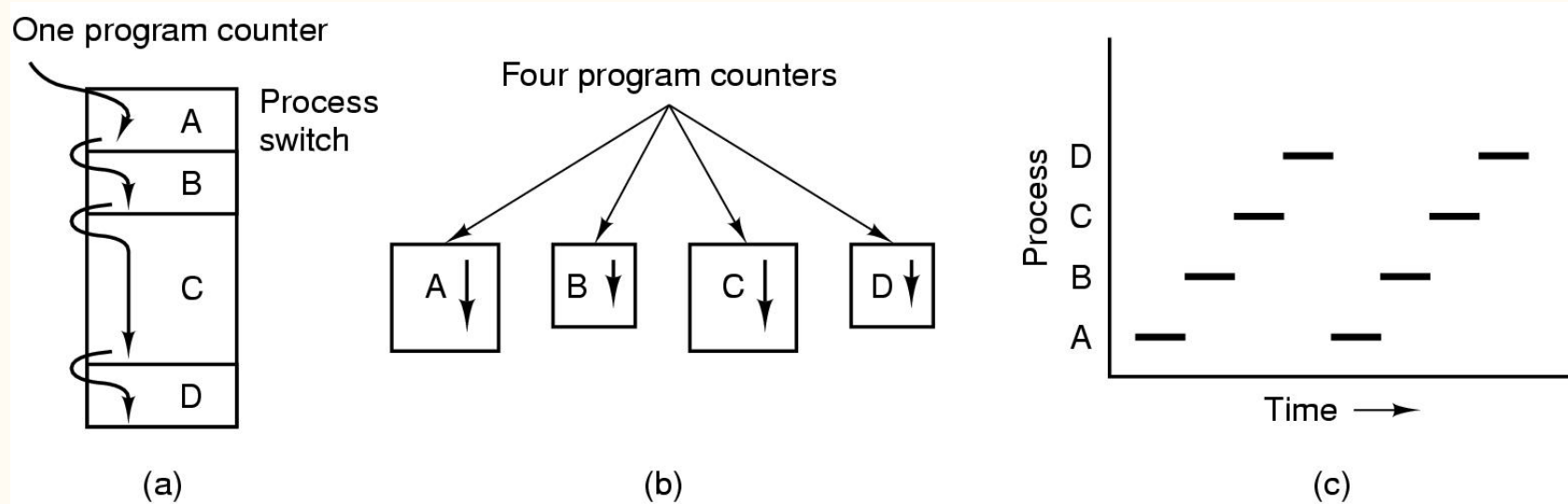
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

Použité obrázky z učebnice:

Andrew. S. Tanenbaum, Structured Computer Organization

<http://www.cs.vu.nl/~ast/books/>

Proces



- Multiprogramming štyroch procesov
- Teoreticky ide o 4 nezávislé procesy
- V skutočnosti ale je v každom okamihu aktívny len jeden

Vznik procesu

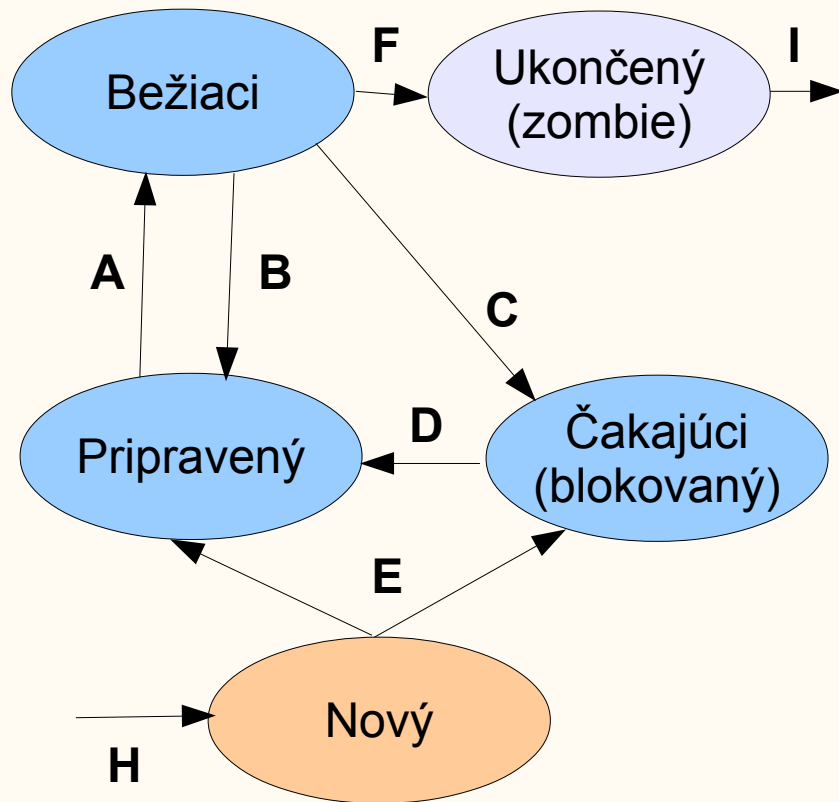
- Pri štarte operačného systému
- Iným procesom - otcovský proces vytvorí synovský proces
 - Pre vykonanie nejakej pomocnej činnosti
 - Ako odozva na akciu používateľa
 - Vykonanie dávky (sálové počítače)

Obvykle otcovský proces odovzdá synovskému procesu nejaké dáta. Najjednoduchšie je to v súbore, ktorého meno sa odovzdá ako parameter v príkazovom riadku.

Zánik procesu

- Ukončí seba (požiada OS o ukončenie s výstupným kódom - `exitCode`)
 - Normálne ukončenie (`exitCode = 0`)
 - Chybové ukončenie (`exitCode != 0`)
- Proces je ukončený operačným systémom
 - Fatálna chyba, v moderných operačných systémoch nastáva keď proces nespracuje vzniknutú výnimku
- Ukončený iným procesom - `kill`

Životný cyklus procesu - stavový diagram



- A** - Plánovač pridelí procesor
- B** - Plánovač odoberie procesor alebo proces sám uvoľní procesor
- C** - Proces vykoná operáciu, súčasťou ktorej je čakanie na nejakú **udalosť**
- D** - Udalosť, na ktorú proces čakal, nastala
- E** - Nové procesy prechádzajú zo stavu Nový do Pripravený alebo Čakajúci
- F** - Procesy končia zo stavu Bežiaci, zostávajú istý čas v stave Ukončený (Zombie)
- H** - Vznik procesu
- I** - Úplný zánik procesu

Vlákná (Threads)

- Procesy sú niekedy príliš ťažkopádne:
 - vznik a zánik je časovo náročný
 - Na komunikáciu treba použiť špeciálne spôsoby (súbory, správy)
- Keď potrebujeme naprogramovať jednu úlohu ako súbor tesne prepojených paralelne bežiacich akcií, hľadáme jednoduchšiu možnosť.
- Riešenie je **vlákno** - na rozdiel od procesu nemá svoju pamäť, ale vlákna jedného procesu ju zdieľajú

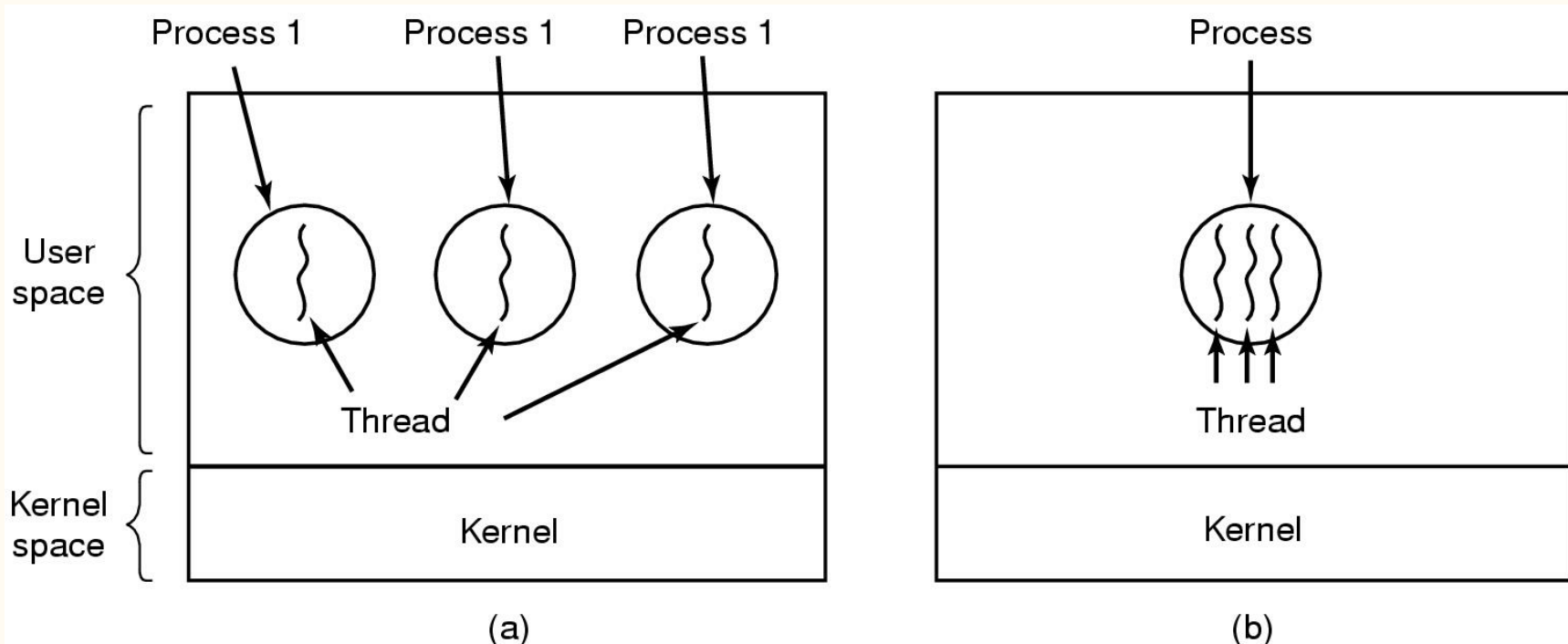
Použitie vlákien

- **Kombinácia rôznych činností** v jednom programe - textový editor reaguje na klávesy, ale „na pozadí“ formátuje a kontroluje pravopis.
- **Serverové aplikácie** - každá prijatá požiadavka sa spracuje v novom vlákne – **worker thread**.
- **Zrýchlenie výpočtov** - keď chceme naplno využiť viacprocesorový systém alebo viacjadrový procesor, tak musíme použiť viac vlákien (toto sa žiaľ nedá ukázať v Pythone, lebo štandardný Python nedokáže naozaj využiť viacjadrové procesory).

Vlákná (Threads)

Vláknó je “odľahčený proces”

Proces robí “obal” pre viacero bežiacich vlákien, ktoré môžu zdieľať pamäť a niektoré stavové informácie procesu.



(a) Tri procesy, každý má jedno vlákno

(b) Jeden proces s tromi vláknami

Vlákná (2)

Proces má

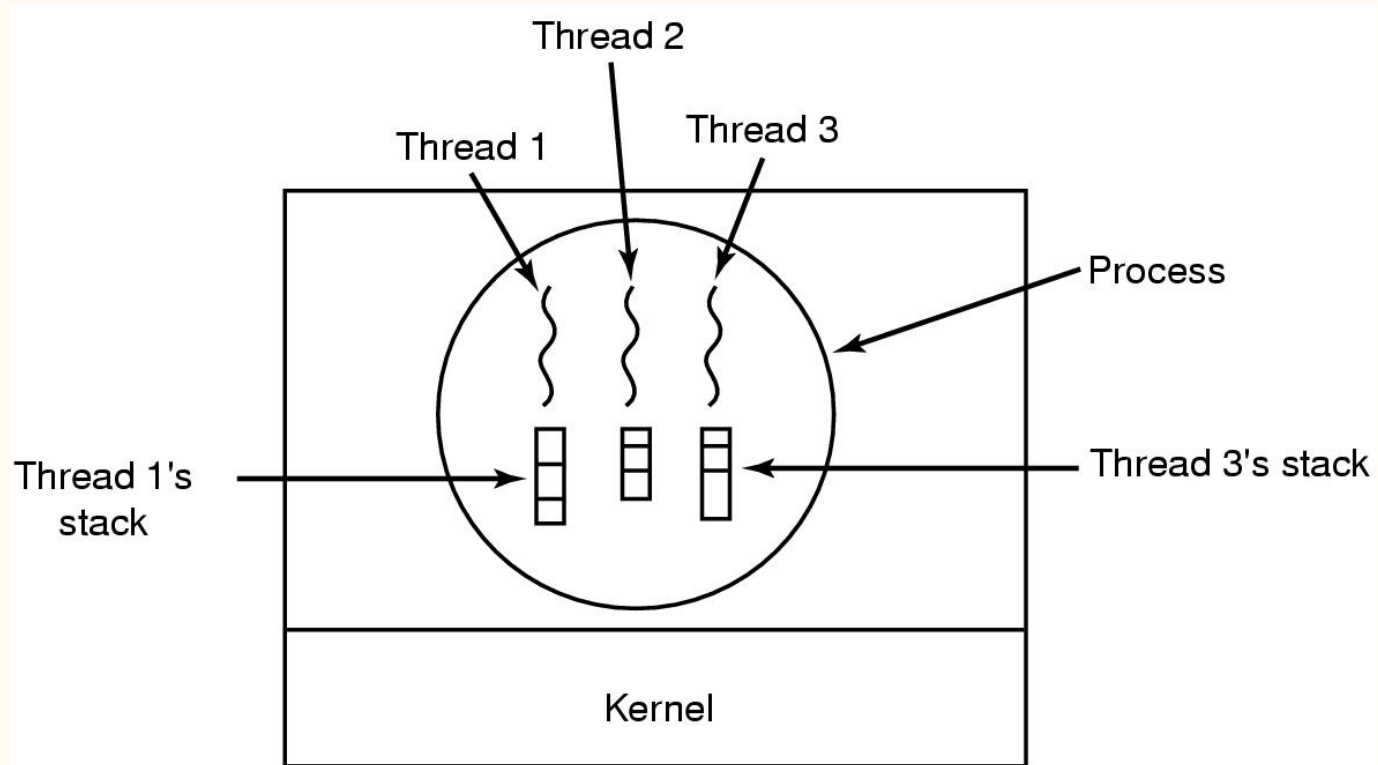
- Virtuálnu pamäť
- Globálne premenné
- Otvorené súbory
- Synovské procesy
- Účtovnú informáciu

Vláknó má

- Program counter
- Obsah registrov
- Zásobník
- Stav threadu

Toto všetko zdieľajú všetky vlákna patriace danému procesu

Vlákna (3)



Každé vlákno má svoj vlastný zásobník

Ukážka – vytvorenie procesu v Pythone

- Naprogramujeme funkciu, ktorá vytvorí ďalší proces, v ktorom vyvolá textový editor Notepad na konkrétny súbor.

```
import subprocess
def edituj(text):
    with open('pom.txt', mode='w') as f:
        f.write(text)
    res=subprocess.run(['notepad', 'pom.txt'])
    if res.returncode == 0:
        for li in open('pom.txt'):
            print(li, end=' ')
    else:
        print('CHYBA', res)
```