Čerstvě hacknuto: Bezpečnostní problémy současných procesorů

Rudolf Marek
SYSGO s.r.o.

Osnova přednášky

- O mně
- Úvod, vývoj hardware
- Útoky na hardware validated boot
- Postranní kanály
  - Útoky využívající postranní kanály
  - Meltdown a Spectre
    - Principy
    - Současný stav
    - Mitigace pro různé procesory

O mně...

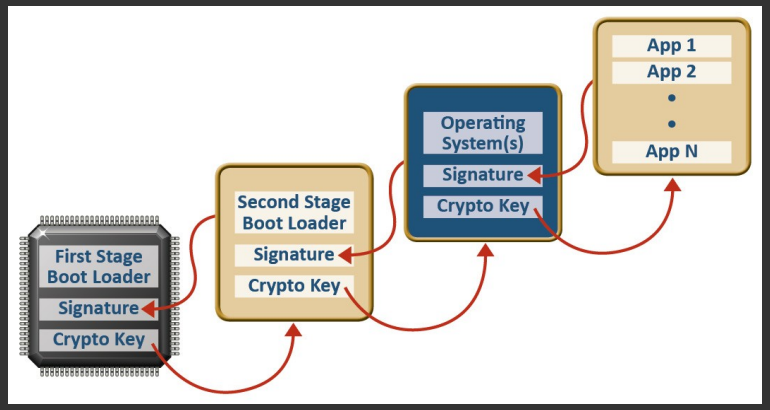
- Pracuji ve společnosti SYSGO s.r.o.
  - R&D centrum v Praze
  - PikeOS – virtualizační realtime mikrokernel
- Opensource příspěvky:
  - Linuxové Kernel Drivery pro HWMON a I2C
  - Opensource BIOS - coreboot
- Hacking
  - Pro radost z poznání
  - hodně low level, hardware, firmware
  - security problémy v x86 procesorech
- Přednáška na valné hromadě Elektrotechnické asociace
  - „Úskalí propojeného světa pro průmyslové aplikace“

Vývoj hardware

- Obrovský růst komplexity hardware
- Hardware se navrhuje “programovatelný”
- Nejen v klasickém software (OS, aplikace) jsou chyby
  - Chyby ve firmware
  - Chyby v hardware

Hardware validated boot

- Hardware nespustí žádný software (bootloder) který není “autentický”
- Řetězové ověřování autenticity (chain of trust)
  - První se spouští speciální ROM v procesoru
  - Každý program ověří další program před spuštěním



Picture from http://www.iconlabs.com/prod/sites/default/files/securebootLR.jpg

Hardware validated boot
Použití

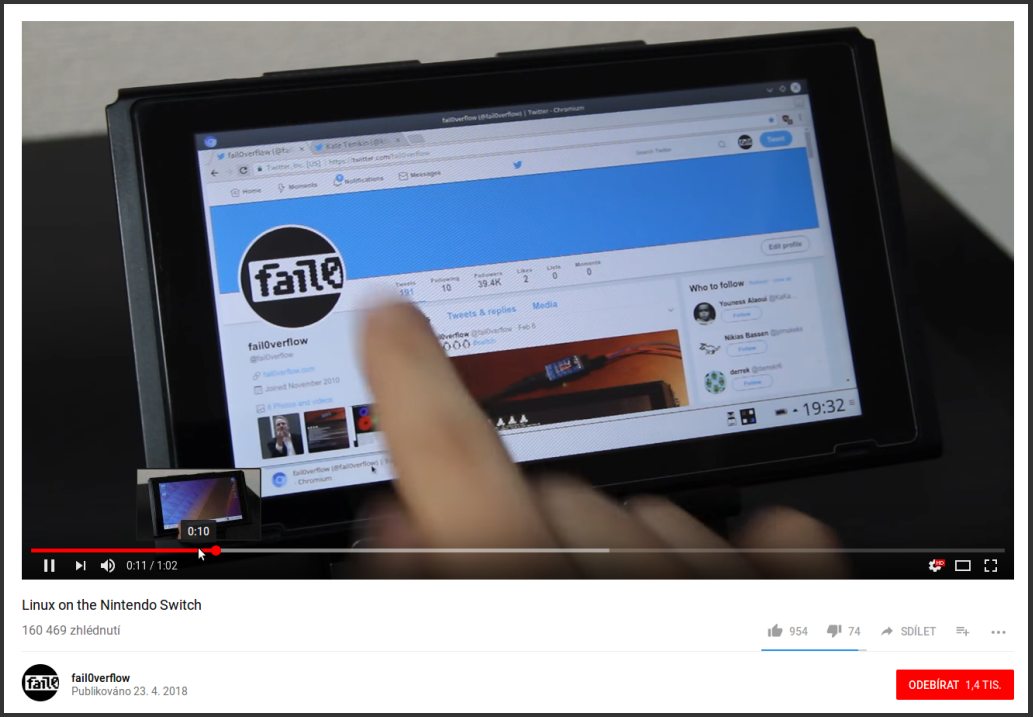
- Zamezí spouštění kódu třetích stran (vendor lock-in)
  - Mobilní telefony, herní konzole...
  - UEFI secureboot
- Ověření autenticity
  - Ověří zda je software na počítači autentický
    - Ochrana před malware
    - Ochrana před zásahy uživatelů, hackerů či státu

Aktuální problémy

- CVE-2018-6242
  - Nvidia Tegra
  - Chyba v bootROM implementaci USB recovery režimu
  - 24.4.2018
- CVE 2017-7932
  - 7.8.2017, chyba bootROM, projde falešný certifikát
  - NXP/Freescale

    i.MX 28 i.MX 50, i.MX 53, i.MX 7Solo i.MX 7Dual Vybrid VF3xx, Vybrid VF5xx, Vybrid VF6xx, i.MX 6ULL, i.MX 6UltraLite, i.MX 6SoloLite, i.MX 6Solo, i.MX 6DualLite, i.MX 6SoloX, i.MX 6Dual, i.MX 6Quad, i.MX 6DualPlus, and i.MX 6QuadPlus

„Ilegální“ - Linux na Nintendu Switch! [0]



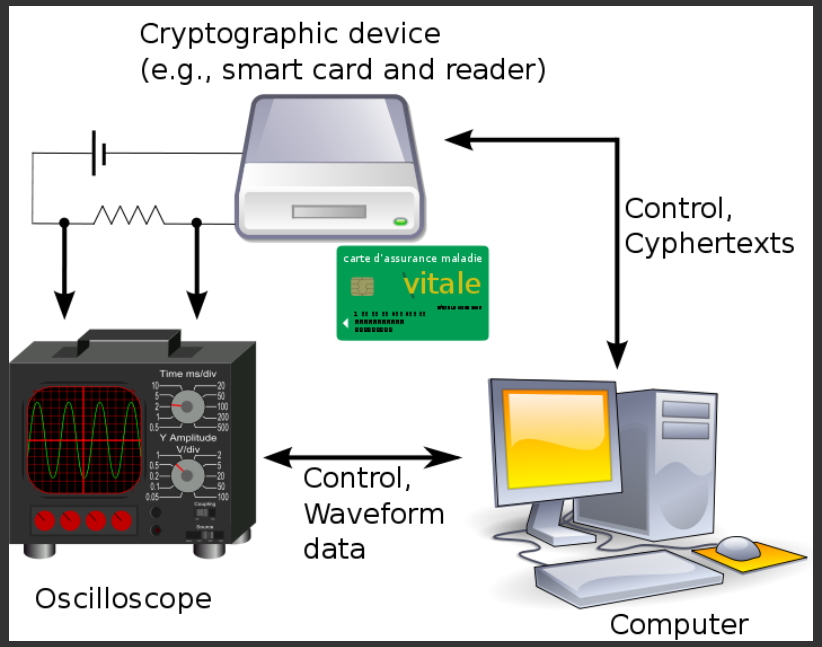
Pictures from www.fail0verflow.com

Útoky postranním kanálem [1]

- Útok využívající extra informace
  - Spotřeba el. energie
  - Čas
  - Elektromagnetické vlny
  - Zvuk
  - Obraz
  - Zbytková data
  - **Stav sdílených částí procesoru**
- Závislý na implementaci
- Nezávislý na algoritmu

Jak získat tajný RSA klíč měřením proudu [2]

- Měřením proudu v čase „okoukáme“ tajný klíč
  - Pokud je v klíči bit 0, mocníme
  - Pokud je v klíči bit 1, mocníme a násobíme



By Audriusa. Recorded by student in ETH (Zurich) during system security laboratory work. - Own work, GPL, https://commons.wikimedia.org/w/index.php?curid=9762292

Postranní útoky s využítím cache

- Cache je „vyrovnávací paměť“
  - rychlost RAM je malá proti rychlosti procesoru
- Využijeme faktu že:
  - CPU cache je sdílená
    - Mezi aplikacemi/jádrem operačního systému
    - Mezi ostatními procesory
    - Nemá kontext procesu/aplikace/jádra OS (na rozdíl od registrů procesoru)
  - Spekulativní spuštění zanechá v cache stopu
    - Spouští se více instrukcí do budoucnosti
  - Trvá rozhodně déle data získat z paměti než z cache

Útok Meltdown CVE-2017-5754 [3]

- Data:
  - Objeven v červnu 2017
  - Zveřejněn v lednu 2018
- Narušuje
  - Isolaci paměti mezi aplikacemi a operačním systémem
  - Útočník dovede
    - číst veškerou paměť v systému
    - Získat tak všechna tajemství
    - Vrací nás do doby DOSu
  - Útočník nedovede
    - Modifikovat data, pouze je číst

Picture from https://spectreattack.com/meltdown.png

Meltdown
Postižené procesory

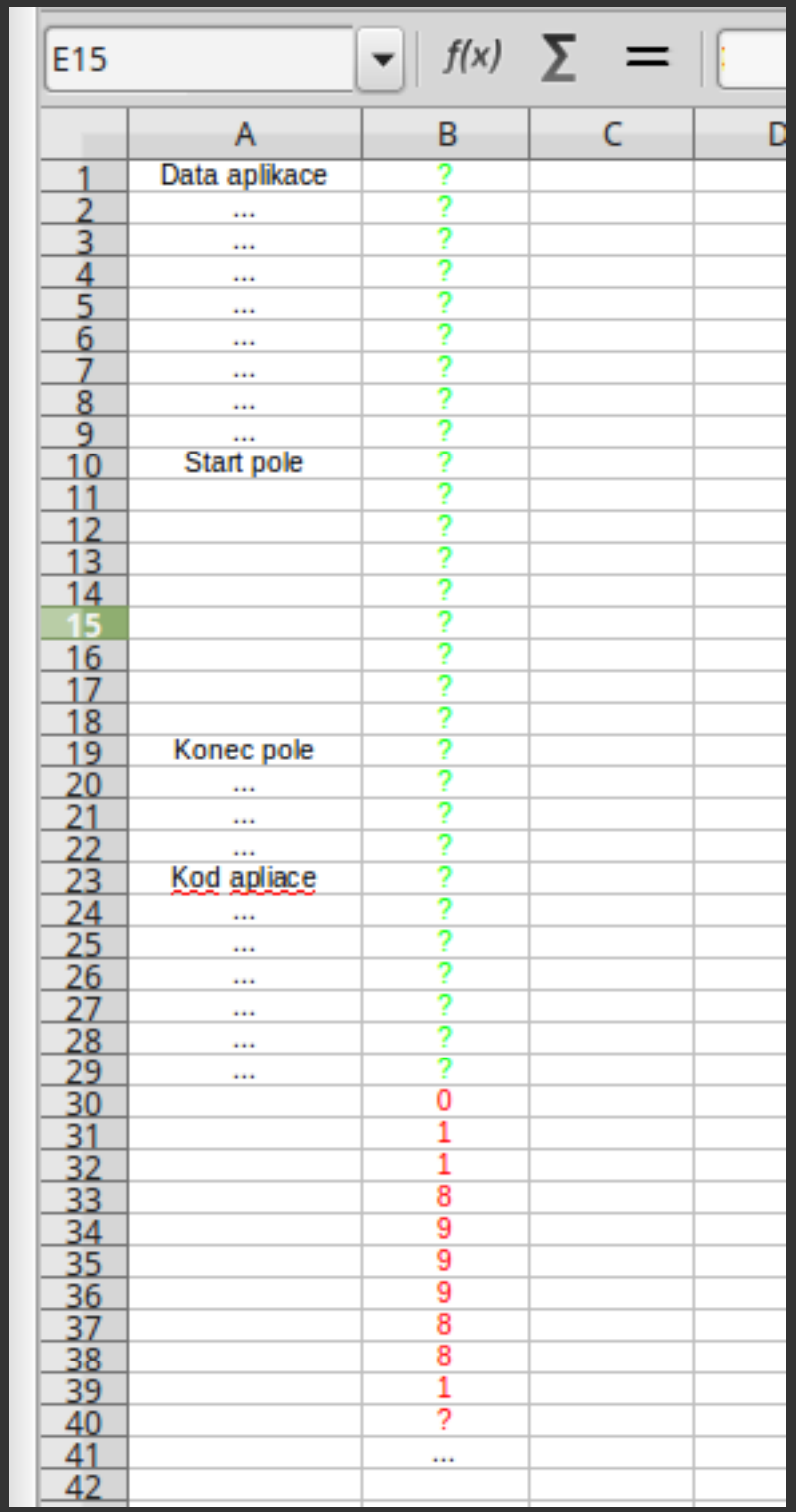
- Procesory x86
  - „Pouze“ procesory Intel
  - Od Intel Pentium (1993) a novější
  - AMD ne
- Procesory ARM
  - Jádra A75
- Procesory PowerPC
  - Informace u NXP na vyžádání pod NDA

Meltdown
Postižené platformy

- Operační systémy
  - Všechny
- Hardwarová virtualizace
  - Ne, průnik mezi virtuálními stroji (VM) není možný
  - Jen v rámci jednoho virtuálního stroje

Meltdown
Princip útoku

- Útočník napíše vlastní program
  - Běží jako běžná aplikace, nemá vyšší práva
  - Vypláchne cache
  - Provede přístup do zakázané oblasti paměti
  - Zakázanými daty indexuje data v poli
  - Podle očekávání program havaruje už při prvním přístupu
  - Ovšem díky spekulativnímu spouštění se změnil stav cache a nyní je můžeme změřit doby přístupu do našeho pole



- Definice: HODNOTA(x): vrátí data na řádku x, sloupce A
- útočník vypláchne z cache řádky A10-A19
- provede následující kód:
  - r = HODNOTA(39)
  - x = HODNOTA(10 + r)
- Změří dobu přístupu do buněk A10 - A19 a zjistí že nejrychleji čteme buňku A11, původní data jsou „1“

Meltdown
Proč funguje

- Honba za rychlejšími a výkonnějšími procesory
- Spekulativně se spustí i další instrukce I když se stav registrů zahodí stav cache ne
- Procesor Intel testuje přístupová práva příliš pozdě, až když se výsledek instrukce promítá v programovém pořadí

Meltdown
Obrana

- Podstatná změna architektury operačního systému
  - Extra isolace adresních prostorů
  - Pokles výkonu asi o 20%

Meltdown varianta 3a

- Útok meltdown lze generalizovat
- Místo čtení ze zakázaných oblastí paměti čteme privilegované registry
- Útočník získá přístup k obsahu privilegovaných registrů
- Původně postihoval jen procesory ARM...

Meltdown 3a
Strikes again!

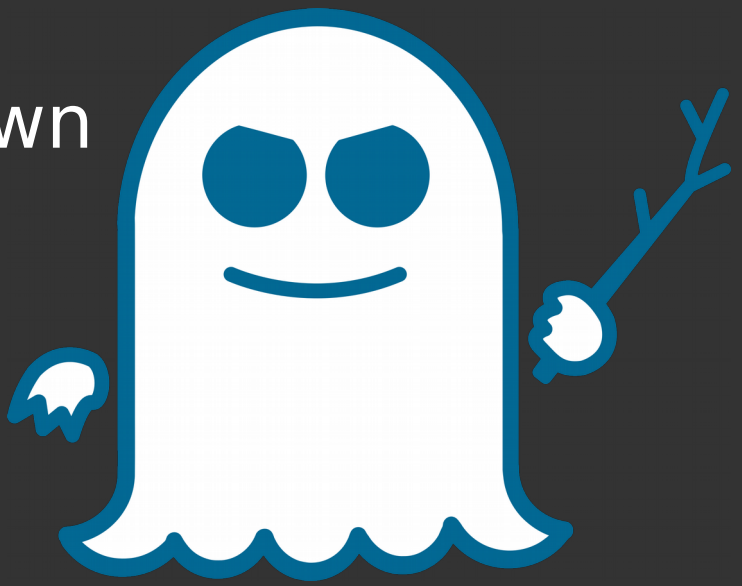
- V SYSGO jsme pečlivě pracovali na záplatách pro Meltdown a Spectre...
- “**Intel** would like to acknowledge and thank Zdenek Sojka, Rudolf Marek and Alex Zuepke from **SYSGO AG** (https://sysgo.com) for reporting CVE-2018-3640.”
- Data:
  - Nahlášení Intelu Leden 2018
  - Zveřejnění 21.5.2018

Meltdown 3a
Postižené procesory

- X86
  - Intel
- ARM – označuje Meltdown jako “variantu 3”
  - Cortex A72, A57, A15
- PowerPC (NXP)
  - Informace u NXP na vyžádání pod NDA

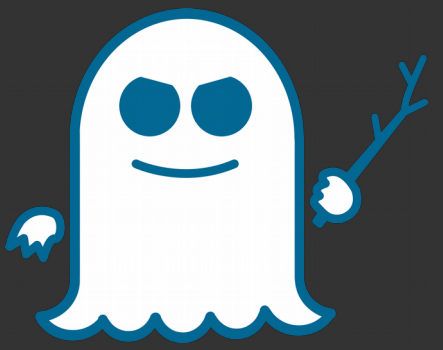
Útoky typu Spectre [4]

- Útok podobný Meltdown
- Využívá také cache jako postranní kanál
- Vyskytuje se ve dvou variantách
  - Varianta 1, oprava software
  - Varianta 2, oprava hardware
- Opět jen čtení dat jako Meltdown
- Problém pro aplikaci i OS
- Využívá predikci skoků



Picture from https://spectreattack.com/spectre.png

Spectre varianta 1
CVE-2017-5753 [4]



- Stejný princip jako Meltdown
  - Donutí aplikaci nebo OS číst jinak nepřístupná data (tajemství)
  - Útok mezi virtuálními stroji
- Mechanismus útoku
  - Zmátne prediktor skoků
  - Aplikace / OS přistoupí k datům co existují ale neměla být přístupná
  - Tyto data se musí použít ještě jednou k jinému přístupu do paměti
  - Zbytek triku funguje podobně jako Meltdown, analýzou stavu cache se dá zjistit co na zakázaném indexu x bylo, který je ovšem mimo pole array1

```
if (x < array1_size) {
   y = array2[array1[x]*256];
}
```

Spectre varianta 2
CVE-2017-5715 [4]

- Zneužívá prediktor nepřímých skoků
- Útok funguje mezi virtuálními stroji
- Útočník je schopen spustit spekulativně libovolný kód
- Pokud najde podobnou sekvenci jako pro variantu 1 může opět číst “tajná” data

Spectre varianta 1 a 2
Postižené procesory

- X86
  - Všechny (AMD i Intel)
- ARM
  - Cortex R7, R8, A8, A9, A12, A15, A17, A57, A72, A73, A75
- PowerPC
  - Informace u NXP na vyžádání pod NDA

Spectre varianta 1
Mitigace

- Nutné zastavit spekulaci při vyhodnocování podmínky
  - Vložením datové závislosti
  - Instrukcí co zastaví spekulaci
    - X86 LFENCE
    - ARM – nová instrukce CSDB

Spectre varianta 2
Mitigace

- X86
  - Změna kompilátoru, kompilátor generuje “retpoline” místo nepřímých skoků
  - Nový mikrokód – IBPB/IBSR/STIB + OS patch
- ARM
  - Invalidace branch prediktoru voláním secure firmware + OS patch

Spectre varianta 4
CVE-2018-3639 [5]

- Data:
  - Objeven ?
  - Zveřejněn 21.5.2018
- Čtení dat ke kterým by neměl být přístup
- Útok využívá vlastnost CPU
  - Speculative store bypass
  - Za jistých okolností procesor dovolí čtení dat před tím než dokončí předchozí zápisy
  - Tak můžeme donutit procesor přečíst spekulativně jinou paměť

Odkazy

- [0] https://fail0verflow.com
- [1] https://en.wikipedia.org/wiki/Side-channel_attack
- [2] https://en.wikipedia.org/wiki/Power_analysis
- [3] https://meltdownattack.com/
- [4] https://spectreattack.com/
- [5] https://www.intel.com/content/www/us/en/security-center/advisory/intel-sa-00115.html

Závěrem

- Pravděpodobně další útoky budou následovat
- Šance pro RISC-V
  - Otevřená specifikace procesoru
  - Otevřená implementace procesoru
- SYSGO
  - PikeOS verze 4.2.1
    - Mitigace pro Meltdown a Spectre
  - Držíme krok s vývojem útoků
  - V rámci vývoje ochran útočíme na vlastní operační systém

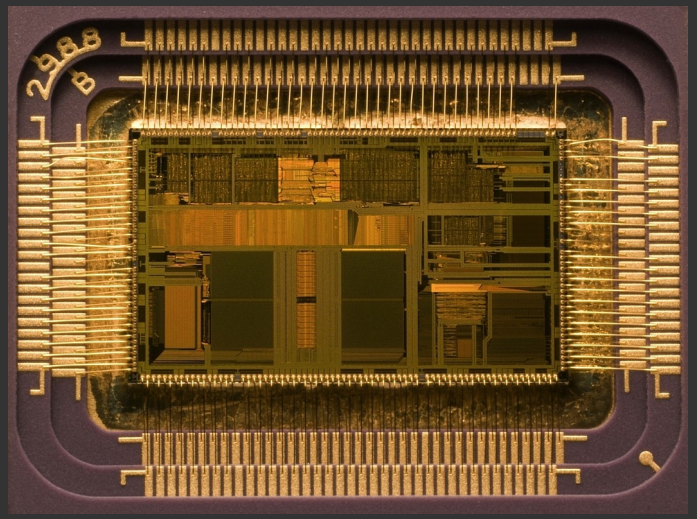
Otázky?

Děkuji za pozornost!

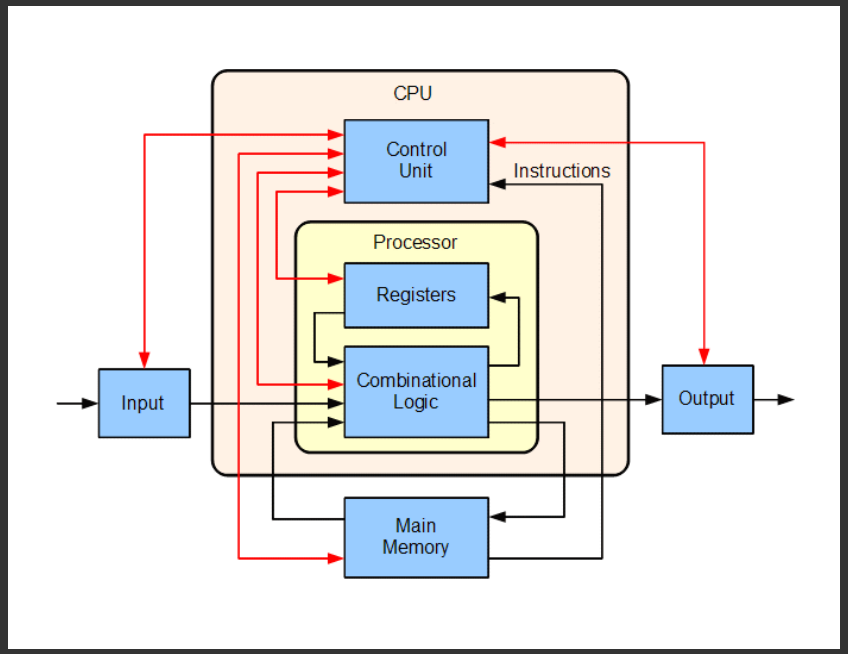
rudolf . marek @ sysgo.com

Co je procesor

- Obsahuje
  - ALU aritmeticko logickou jednotu
  - Registry (malá pracovní paměť)
  - Cache (vyrovnávací paměť)



CC-BY-SA 3.0 Intel 80486DX2 – from Uhbernguin



By Lambtron - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=37716438

Procesor a program

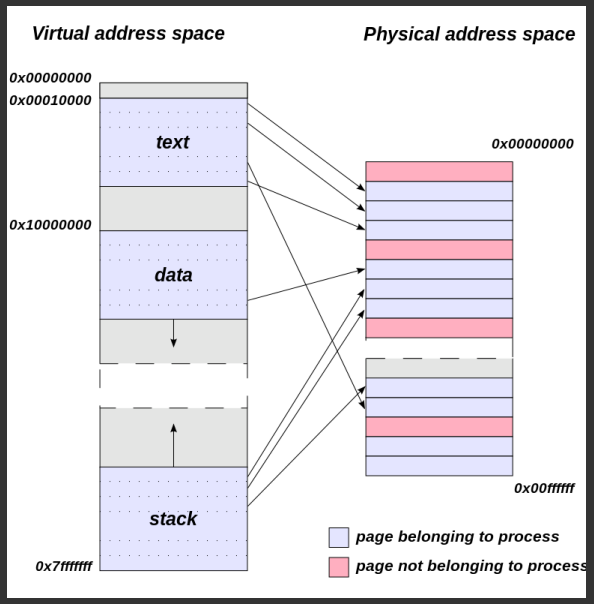
- Program = algoritmus + data
  - Procesor
    - Zpracovává instrukce, načítají se z paměti programu
    - Provádí aritmetické operace, logické operace
    - Načítá data z paměti do registrů a naopak
    - Podmíněné a nepodmíněné skoky, mění tok programu
  - Superskalární procesor
    - Zpracovává instrukce mimo pořadí programu
    - Ale efekty se promítají v programovém pořadí (skoro)

Co je cache

- Princip lokality a času
  - Data blízko sebe budou potřeba
  - Data co jsou potřeba teď budou potřeba za chvíli
  - Velikosti cache řádově KiB, MiB
- Vyrovnávací pamět
  - Maskuje rozdílou rychlost přístupu do RAM a do registrů
  - faktor rozdílu registr / RAM cca 100x
  - Skrytá pro programátora
  - Různé úrovně L1, L2, L3
  - Pracuje s fyzickými adresami
  - **Důsledek**: některá data jsou dostupná rychleji

Co je virtuální adresní prostor procesoru

- Množina všech adres
  - Část z nich využívá aplikace
    - Různé aplikace mají stejné adresy
    - Aplikace se navzájem „nevidí“
  - Část jádro systému
    - Aplikace sem nemají přístup
- Virtuální adresy jsou překládány na fyzické
  - Překlad zajišťuje hardware
  - Tabulky překladu spravuje operační systém



Pictures from https://en.wikipedia.org/wiki/Virtual_address_space