Univerzita Hradec Králové
Fakulta informatiky a managementu

Správa paměti

Mgr. Josef Horálek

FIM UHK

Správa paměti

- Jeden z hlavních úkolů jádra systému – zajistit dostatek prostoru pro jádro a všechny procesy při omezené velikosti paměti a co nejmenším zdržením režijními operacemi

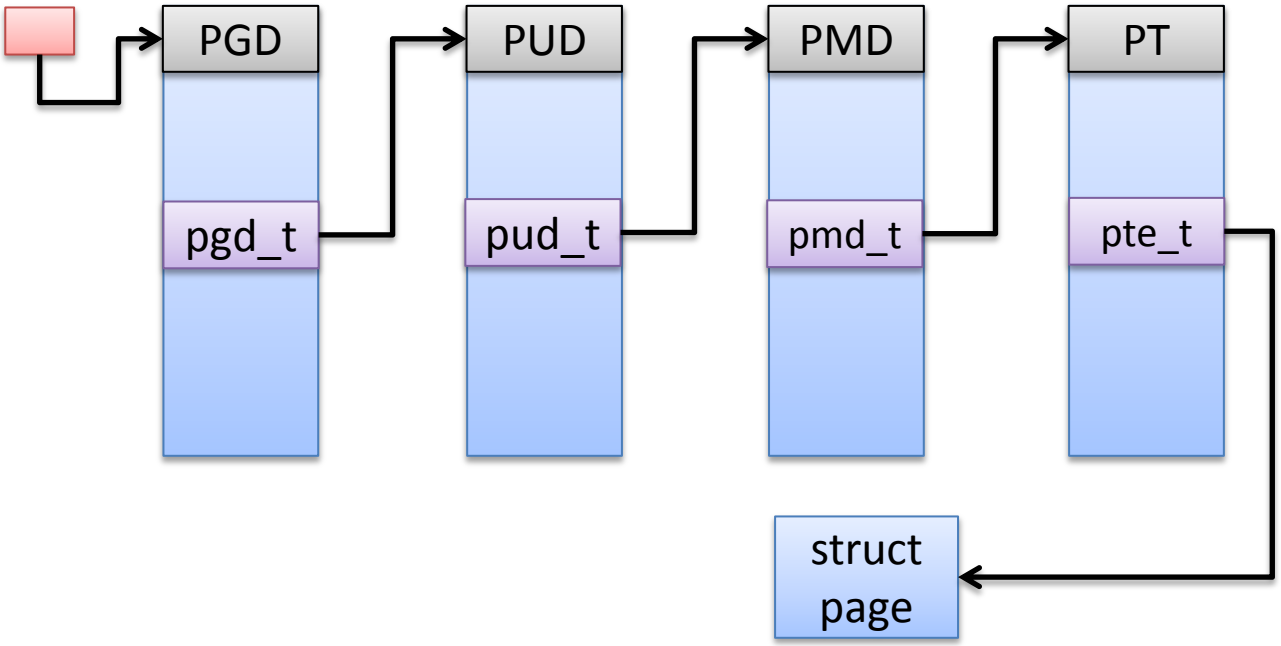
FIM UHK

Správa paměti – hlavní úkoly

- Převody mezi různými druhy adres
- Zpřístupnění paměti, která leží mimo adresní prostor
- Nízkoúrovňová alokace pro různé účely
- Správa paměti a alokace pro jádro
- Správa paměti procesů, přidělování, mapování, sdílení
- Různé cache, přednačítání, zpožděný zápis
- Odkládání na disk, zpětné načítání
- Řešení nedostatku paměti

FIM UHK

Stránkování paměti



- PGD – page global directory – globální adresář stránek
- PUD – page upper directory – horní adresář stránek
- PMD – page middle directory – střední adresář stránek
- PT – page table – tabulka stránek

FIM UHK

Zóny paměti

- Z důvodů HW limitů je paměť rozdělena na zóny, kde každá má své vlastnosti předurčující její použití
- Dělení na zóny je dáno architekturou – liší se u různých architektur – obecně:
  - Zóna pro DMA
  - Zóna pro normální použití
  - Zóna mimo adresní prostor

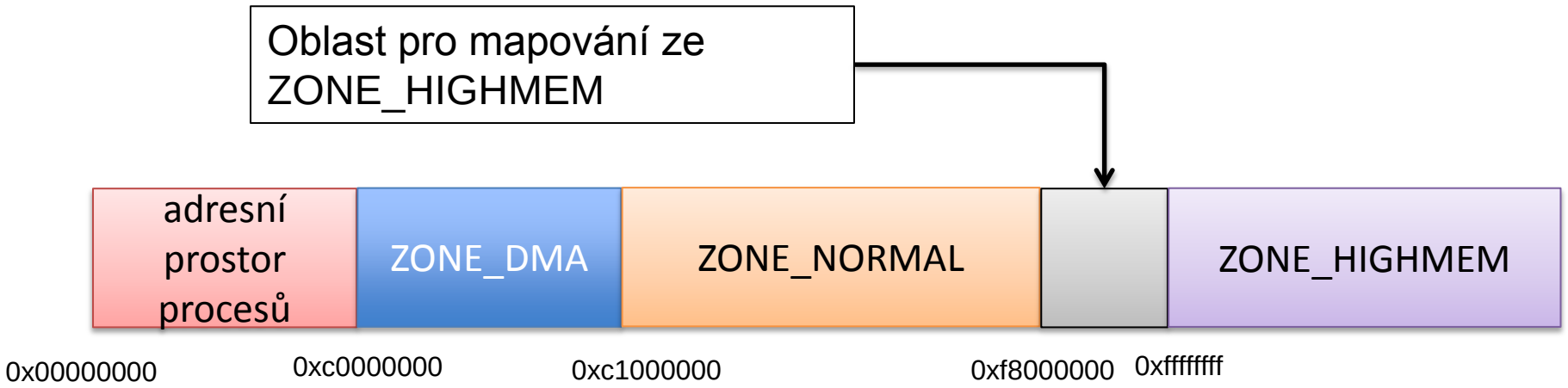
FIM UHK

Zóny paměti

FIM UHK

Zóny na architektuře x86

- Nejčastější problémy dané HW limity 32bitové architektury x86



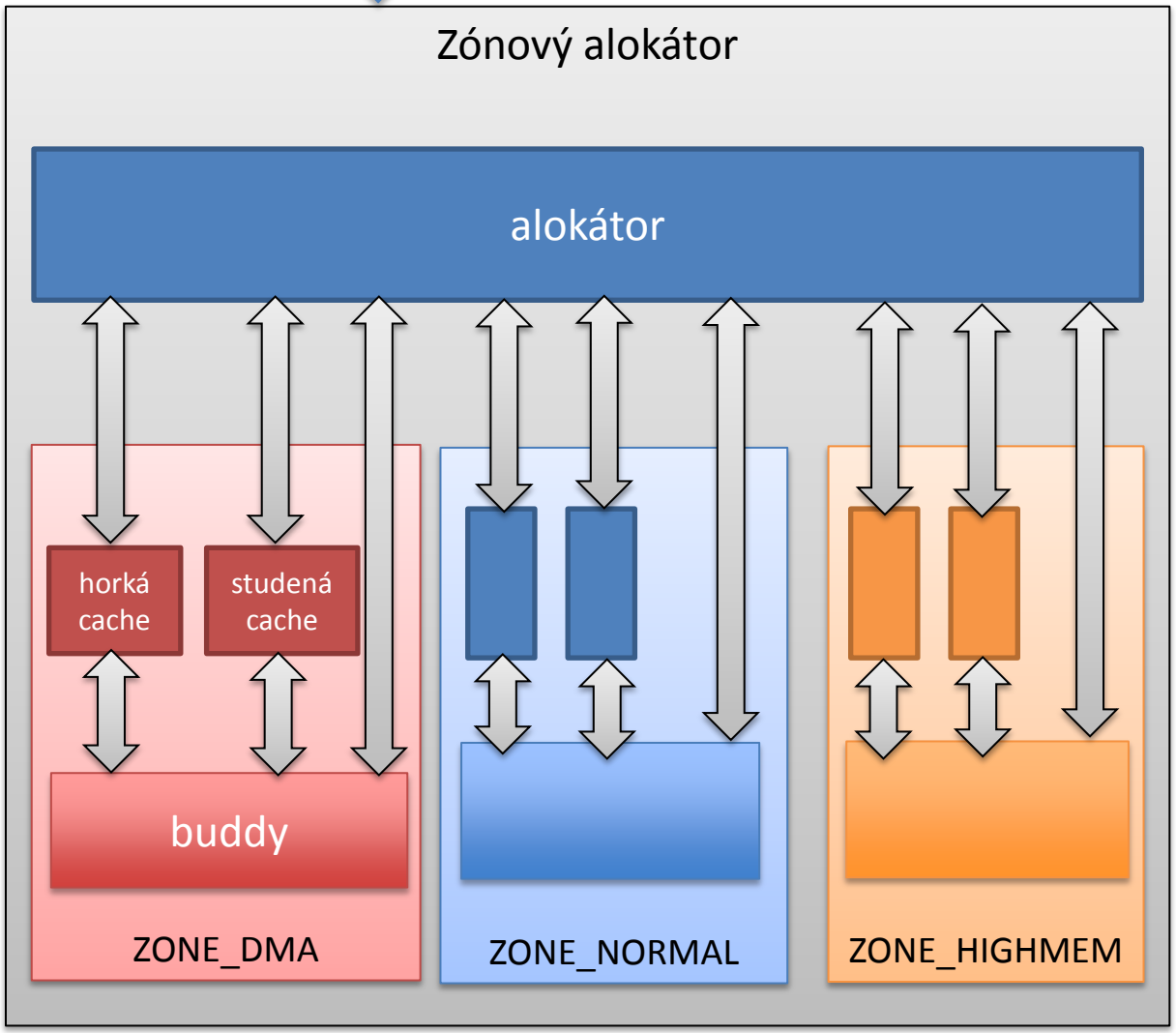
FIM UHK

Zóny na architektuře x86_64

- 64bitový adresní prostor - zjednodušení situace
- Přímo lze adresovat oblast 64 EB – nevyužívá se tedy zóna horní paměti

FIM UHK

Alokace stránek



FIM UHK

Algoritmus buddy

- Skupiny bloků od 1 do 1024 stránek
- Vždy zdvojnásobuje další úrovně velikosti
- Celkem 11 úrovní velikosti
- Volné bloky do seznamů (pro každou úroveň zvlášť)

FIM UHK

Algoritmus buddy

- Pravidla pro alokaci:
  - Zkus najít blok odpovídající velikosti – pokud existuje – přiděl ho
  - Zkus najít blok o úroveň větší velikosti – neexistuje-li pokračuj na další úroveň
  - Nalezený blok rozděl tak, aby se přidělila potřebná část a zbytek byl použitelný pro další zpracování

FIM UHK

Algoritmus buddy

- Při uvolňování se postupuje opačně. Lze-li sousední dva bloky sloučit učiní se tak. Postupuje se rekurzivně tak dlouho, dokud lze slučovat.

FIM UHK

Rezervované stránky

- Normální alokace – není-li k dispozici volný blok paměti – volání se uspí – spuštěn mechanismus uvolnění paměti – uvolní dostatečnou velikost – alokace probuzena
- Ne vždy možné – jádro udržuje některé stránky vyhrazené pro použití, kdy se alokace nesmí blokovat
- Tato paměť se dělí mezi ZONE_NORMAL a ZONE_DMA

FIM UHK

Cache pro stránky

- Vylepšení výkonnosti při alokaci
- Často se alokují jednotlivé stránky – využití buddy pomalé
- Pro každý procesor v jádře vytvořeny cache – horká (hot cache) a studená (cold cache)

FIM UHK

Proces alokace

- **_alloc_page()** základní alokační funkce volaná z **alloc_pages()**
- Na začátek zkusí získat volné stránky pomocí **get_page_from_freelist()** – pokud ne – volán démon **kswapd**, aby uvolnil dostatečný počet stránek
- Paralelně s démonem proveden druhý pokus o získání stránek **get_page_from_freelist()**
- Pokud ne – proveden třetí pokus

FIM UHK

Proces alokace

- Pokud dojde k selhání i nyní – zacyklení a pokus neustále opakován dokud se nedosáhne úspěchu
- Automatická alokace – zde končí neúspěchem

FIM UHK

Proces alokace

- Normální alokace – dá prostor k běhu jiným procesům – pak synchronní ulovňování paměti **(try_to_free_pages())** – nový pokus získat volné stránky – je-li neúspěšné a současně neuspělo ani synchronní uvolňování = out of memory (nedostatek paměti – funkcí out_of_memory())

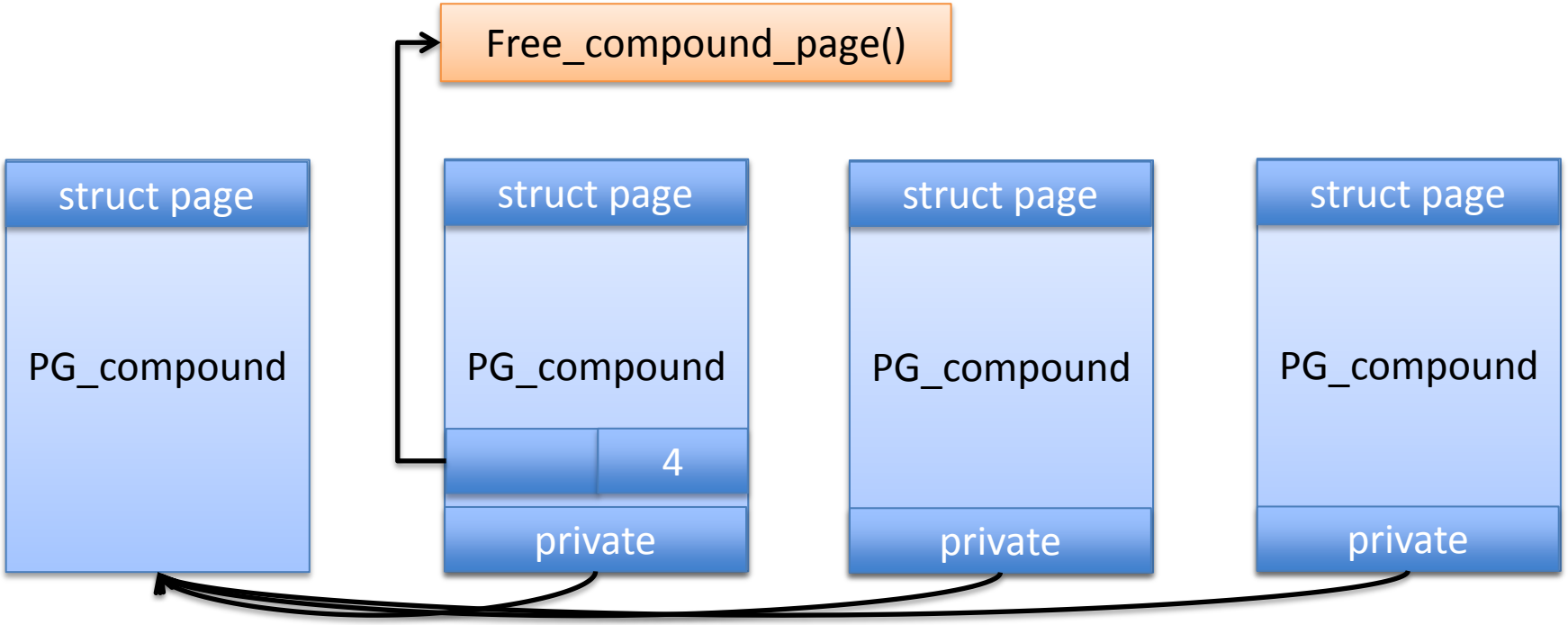
FIM UHK

get_page_from_freelist()

- Funkce prochází jednotlivé paměťové zóny
- Podle použitého limitu rozhoduje o možnosti použití příslušné zóny
- Lze-li zónu použít volá se pro ni **buffered_rmqueue()** pro získání volných stránek, které pracují přímo se zónovým alokátorem

FIM UHK

Sloučené stránky



FIM UHK

Uvolnění stránek

- Jednoduchý proces na vyšší úrovní – složitější na úrovní algoritmu buddy
- **Funkce _free_pages()** dekrementuje počítadlo referencí – testuje uvolnění stránky – volá **free_hot_page()**, která stránku vloží do hot cache aktuálního procesoru – pokud dosažena horní mez cache – funkce **free_pages_bulk()** nadbytečné stránky vrací do buddy seznamů

FIM UHK

Alokace nesouvislých úseků

- Není jednoduché nalézt souvislý úsek paměťových stránek – alokace nesouvislých úseků a jejich namapování
- Alokuje se sada úseků a ty se namapují do virtuálního adresného prostoru jádra – získáme souvisle adresovatelný úsek – pak lze normálně pracovat

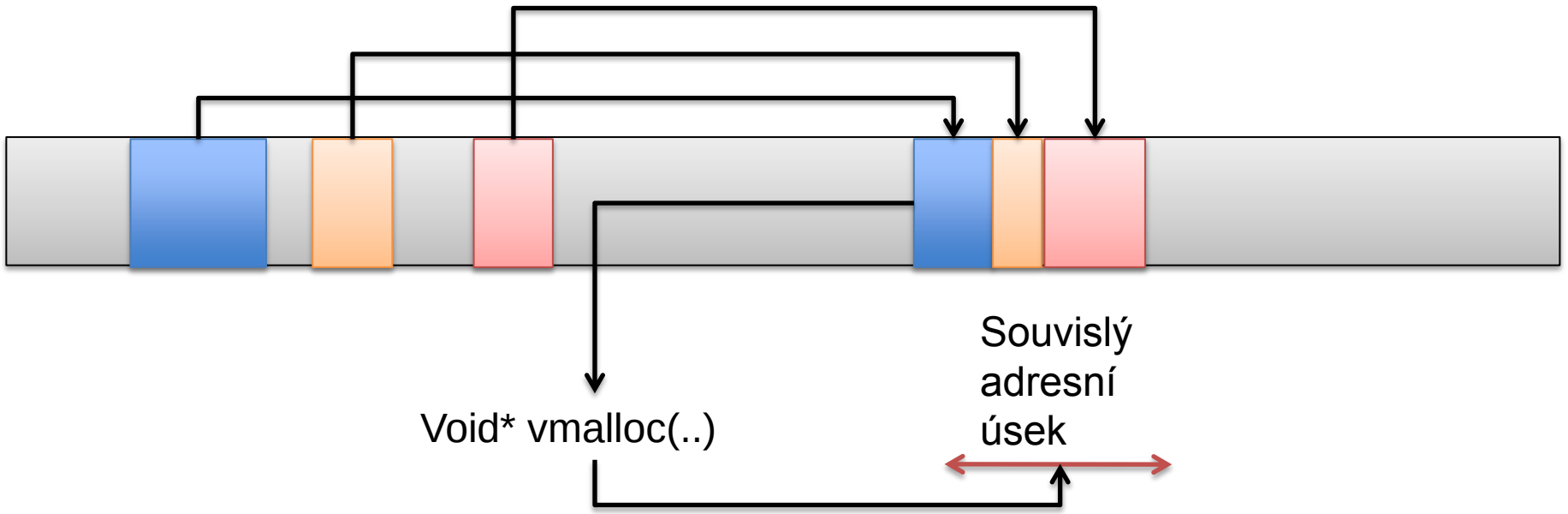
FIM UHK

Alokace nesouvislých úseků

- Funkce **get_vm_area_node()** hledá volnou oblast, kam bude paměť po alokaci namapována, v rozsahu adresního sprostoru vyhrazeného pro tento účel
- Oblast musí být dostatečně velká (velikost + 1stránka navíc – ochrana)
- Vlastní alokace – funkce **_vmalloc_area_node()** - alokuje jednotlivé stránky.
- Ukazatele alokovaných stránek ukládány do pole – použito při mapování

FIM UHK

Alokace nesouvislých úseků



FIM UHK

Uvolnění alokované paměti

- Alokovanou paměť je potřeba po použití uvolnit
- Funkce **vfree()** – volá funkci _**vumap()**, která se stará o odmapování a uvolnění
- Nejdříve dojde k odmapování a je-li potřeba postupně prochází všechny stránky a nastává dealokace

FIM UHK

Řešení nedostatku paměti

- Metody řešení nedostatku fyzické paměti:
  - Uvolnění aktuálně nevyužívaných stránek
  - Aktivace mechanismu OOM killer – postupně ukončí tolik procesů, aby vyřešil situaci
- Kategorie uvolňovaných stránek:
  - Odložitelné
  - Synchronizovatelné
  - Přímo uvolnitelné

FIM UHK

Reverzní mapování

- Používá se pro najití všech položek odkazujících na určitou stránku
- V linuxovém jádře se používá objektové reverzní mapování a to buď anonymní nebo souborové

FIM UHK

mechanismus

- Nejlépe a nejrychleji řešit nedostatek paměti, zároveň nejméně omezovat jádro a procesy
- Pravidla:
  - Přednost mají nejdéle nepoužívané stránky
  - Přednost mají stránky, které nikdo nepoužívá. Stránky, které nepoužívá žádný proces by se měly uvolňovat dříve než stránky použité v procesech
  - Co nejvíc stránek procesů musí být uvolnitelný – všechny stránky kromě aktivních
  - Pro uvolnění sdílených stránek se musí odmapovat všechny odkazující položky

FIM UHK

OOM killer

- Situace, kdy se nedaří uvolnit žádnou další paměť, kterou je potřeba alokovat – násilné ukončení procesů
- Začíná se výběrem vhodného kandidáta na ukončení – hledá funkce **select_bad_process()** – prochází všechny procesy a vybere ten s nejvyšším skóre (bedness)

FIM UHK

Výpočet badness() - ovlivňuje

- Proces **swapoff** – systém si nemůže dovolit přijít o swap – maximální skóre
- Alokovaná paměť procesu – přímá hodnota
- Počet potomků – ½ paměti každého potomka rodiči
- Procesorový čas – skóre dělíme druhou mocninou času procesoru v desítkách sekund

FIM UHK

Výpočet badness() - ovlivňuje

- Reálná čas – skóre se dělí čtvrtou odmocninou času běhu procesu v desítkám sekund
- Hodnota nice – je li >0, skóre se zdvojnásobí
- Administrátorské procesy – procesům s UID=0 nebo EUID=0 a procesům CAP_SYS_ADMIN se skóre dělí čtyřmi
- Procesy s přímím přístupem k HW – procesům WIO se skóre dělí 8

FIM UHK

Výpočet badness() - ovlivňuje

- Společné uzly – pokud proces nemá s aktuálním procesem společné paměťové uzly, skóre se dělí osmi
- Nastavená hodnota – hodnota **oom_adj** říká, o kolik se má skóre posunout – tedy hodnota 3 je posun o 3 bity – osmina původního skóre

FIM UHK

OOM killer

- Nenalezne-li funkce žádného kandidáta, je systém zastaven, protože nelze pokračovat dál.
- Pokud vhodný proces nalezen – volá se funkce **oom_kill_process()** – začne testem zda proces již není ukončován – pokud ano nastaví se mu příznak **TIF_MEMDIE** a nic dalšího se neprovede

FIM UHK

OOM killer

- Než začne ukončovat přímo vybraný proces pokusí se ukončit některého z jeho potomků
- Pokud úspěšné – nepokračuje
- K ukončení používána funkce **oom_kill_task()** – ta ukončí určitý proces a současně i všechny ostatní procesy, které s ním sdílejí paměťový kontext.
- Samostatné ukončení se provádí zasláním signálu **SIGKILL**

Univerzita Hradec Králové
Fakulta informatiky a managementu

Děkuji za pozornost…