

Struktura jąder systemów operacyjnych

Lista zadań nr 7

Na zajęcia 20 maja 2020

UWAGA! W trakcie prezentacji rozwiązań należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **wytluszczoną** czcionką.

Zadanie 1. Systemy uniksowe prezentują użytkownikowi **średnie obciążenie** (ang. *load average*) systemu w postaci trójki liczb obliczanych z użyciem **wykładniczego uśredniania** (ang. *exponential averaging*). Przeczytaj procedurę `loadavg` i podaj wzór na wyznaczanie średniego obciążenia.

Wskazówka: Przeczytaj najpierw ustęp „Shortest Process Next” z [2, §9.2].

Zadanie 2. Na podstawie [2, §9.3] opisz algorytm **planisty długoterminowego** stosowanego we wcześniejszych systemach uniksowych. Uwzględnij **kolejki wątków** i sposób wyznaczania priorytetów. Na podstawie komentarzy dotyczących «`l_estcpu`» w linii 190 wyjaśnij sposób wyznaczania zużycia procesora.

Zadanie 3. Przeczytaj [2, §10.3] lub [1, §10.3.4] o algorytmie Completely Fair Scheduler stosowanym w bieżących wersjach jądra Linux. Opowiedz o niedostatkach planisty $O(1)$ i implementacji idei **sprawiedliwego szeregowania** (ang. *fair-share scheduling*). W jaki sposób algorytm CFS wybiera kolejne zadanie do uruchomienia? W jaki sposób nalicza wirtualny czas wykonania «`vruntime`», aby wymusić sprawiedliwy przydział czasu procesora?

Wskazówka: Szczegóły można znaleźć w artykule „[Inside the Linux 2.6 Completely Fair Scheduler](#)”¹ oraz [4, strona 46].

UWAGA! W sieci Internet i [2] można się natknąć na opisy algorytmu ULE w jego pierwszej wersji. Nas obchodzi druga wersja, więc najlepiej ograniczyć się do podręcznika [3] i artykułu pt. [Scheduler ULE](#)².

Zadanie 4. Na podstawie procedury `sched_interact_score` podaj wzór na **stopień interaktywności**, który wyliczany jest na podstawie dwóch parametrów: czasu **przebywania na procesorze** i czasu **snu dobrowolnego**. Omów algorytm `sched_interact_update` zarządzający historią wcześniej wymienionych parametrów. Jaki efekt przynosi przeliczanie wartości «`ts_runtime`» i «`ts_slptime`» w taki sposób?

Zadanie 5. Wyjaśnij jak `sched_priority` wyznacza priorytet wątku na podstawie stopnia interaktywności i priorytetu nadanego przez użytkownika `setpriority(2)`. Następnie przeczytaj procedurę `tdq_runq_add` i powiedz kiedy wątki są umieszczane w kolejce zadań: interaktywnych, z podziałem czasu, jałowych. Dlaczego zadania z podziałem czasu dodajemy do kolejki przy pomocy `runq_add_pri` zamiast «`runq_add`», a usuwamy przy pomocy `runq_choose_from` zamiast «`runq_choose`»?

Zadanie 6. Procedura `sched_pickcpu` służy do wyznaczania docelowego procesora w trakcie dodawania zadania do kolejek planisty ULE. Na podstawie kodu procedury i [3, strona 154 i 155] omów algorytm wyznaczania odpowiedniego procesora – uzasadnij celowość użytych kryteriów. Na jakiej podstawie wyznaczany jest koszt migracji na inny procesor? Pomiń omawianie procedury «`cpu_search`»!

Zadanie 7. Długoterminowy algorytm **równoważenia obciążenia** został omówiony na [3, stronie 156]. Co ile czasu uruchamiana jest procedura `sched_balance`? Przeprowadź uczestników zajęć przez procedurę `sched_balance_pair`, której zadaniem jest zrównoważyć obciążenie między najbardziej i najmniej zajęтым procesorem. Ile zadań na raz przenosi procedura `tdq_move`? Kiedy `tdq_notify` musi wystąpić przerwanie międzyprocesorowe do procesora biorcy?

¹<https://www.ibm.com/developerworks/library/l-completely-fair-scheduler/>

Literatura

- [1] „Modern Operating Systems”
Andrew S. Tanenbaum, Herbert Bos;
Pearson; 4th edition, 2015
- [2] „Operating Systems Internals and Design Principles”
William Stallings; Pearson; 9th edition, 2018
- [3] „The Design and Implementation of the FreeBSD® Operating System”
Marshall Kirk McKusick, George V. Neville-Neil, Robert N.M. Watson;
Addison-Wesley Professional; 2nd edition, 2014
- [4] „Linux Kernel Development”
Robert Love; Addison-Wesley; 3rd edition, 2010
- [5] „Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture”
Richard McDougall, Jim Mauro; Prentice Hall; 2nd edition, 2006
- [6] „FreeBSD Device Drivers: A Guide for the Intrepid”
Joseph Kong; No Starch Press; 2012
- [7] „UNIX Internals: The New Frontiers”
Uresh Vahalia; Prentice Hall; 1996
- [8] „Mac OS X and iOS Internals: To the Apple's Core”
Jonathan Levin; Wrox; 2012