

Systemy operacyjne (zaawansowane)

Lista zadań nr 4

Na zajęcia 9 listopada 2017

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Tanenbaum (wydanie czwarte): 2.2, 11.4
- Stallings (wydanie siódme): 4.1 – 4.3

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

Zadanie 1. Czym różni się **przetwarzanie równoległe** (ang. *parallel*) od **przetwarzania współbieżnego** (ang. *concurrent*)? Czym charakteryzują się **funkcje wielobieżne** (ang. *reentrant*)? Czy funkcje wielobieżne są **wątkowo-bezpieczne** (ang. *thread-safe*)? Swoją wypowiedź podeprzyj fragmentami pseudokodu. Czy w jednowątkowym procesie uniksowym może wystąpić współbieżność?

Zadanie 2. Wymień przewagi **wątków przestrzeni jądra** (ang. *kernel-level threads*) nad **wątkami przestrzeni użytkownika** (ang. *user-level threads*). Czemu biblioteka ULT musi kompensować brak wsparcia jądra z użyciem **opakowań funkcji** (ang. *wrapper*)? Na podstawie dokumentacji biblioteki **GNU pthread**¹ (*POSIX Replacement API*) pogrupuj opakowania funkcji ze względu na pełnione zadania.

Zadanie 3. Opisz hybrydowy model wątków bazujący na **aktywacjach planisty** i pokaż, że może on łączyć zalety KLT i ULT. Posługując się artykułem **An Implementation of Scheduler Activations on the NetBSD Operating System**² wyjaśnij jakie **wezwania** (ang. *upcall*) dostanie planista przestrzeni użytkownika gdy: zwiększymy lub zmniejszymy liczbę **wirtualnych procesorów**, wątek zostanie zablokowany lub odblokowany w jądrze, proces otrzyma sygnał.

Zadanie 4. Na wykładzie mówiliśmy o tym, że w jednowątkowym procesie można zrealizować przetwarzanie współbieżne wielu podprogramów. Warunkiem koniecznym było korzystanie z **nieblokujących** wywołań systemowych. Na podstawie **notatek**³ do rozdziału 6 książki „Unix Network programming, Volume 1” opisz jak przy pomocy techniki zwanej **multipleksowaniem wejścia-wyjścia** (ang. *I/O multiplexing*) zbudować jednowątkowy serwer obsługujący współbieżnie wiele połączeń sieciowych TCP. Wybierz tylko jedno z wywołań **select**⁴ lub **poll**⁵.

Zadanie 5. Najpowszechniej implementowane wątki przestrzeni jądra wprowadzają do programów dodatkowy stopień złożoności. Co dziwnego może się wydarzyć w wielowątkowym procesie gdy:

- wołamy funkcję **fork**, aby utworzyć podproces,
- użytkownik przerywa program z klawiatury wysyłając sygnał SIGINT do procesu,
- korzystamy w wielu wątkach z jednego pliku,
- modyfikujemy zmienną środowiskową funkcją **putenv(3)**⁶,
- wątki intensywnie korzystają z menadżera pamięci z użyciem funkcji **malloc**.

¹<https://www.gnu.org/software/pthread/pthread-manual.html>

²<http://web.mit.edu/nathanw/www/usenix/freenix-sa/freenix-sa.html>

³<https://notes.shichao.io/unp/ch6/>

⁴<http://man7.org/linux/man-pages/man2/select.2.html>

⁵<http://man7.org/linux/man-pages/man2/poll.2.html>

⁶<http://man7.org/linux/man-pages/man3/putenv.3.html>

Zadanie 6. Wątki nie są panaceum na problemy z wydajnością oprogramowania na **maszynach wieloprocessorowych ze współdzieloną pamięcią** (ang. *Shared Memory Processors*). Wymień warunki jakie musi spełniać architektura programu by stosowanie wątków było uzasadnione (§4.3)? Co ogranicza wydajność programów używających wątków? Jakie problemy z efektywnym użyciem wątków napotkali twórcy silnika gry **Half-Life 2**⁷ i jak je rozwiązali?

Zadanie 7. Opowiedz ogólnie o niebezpieczeństwach jakie niesie ze sobą stosowanie wątków. Zapoznaj się z następującymi materiałami:

- [The Art of Unix Programming: Threads – Threat or Menace?](#)⁸
- [Why Threads Are A Bad Idea \(for most purposes\)](#)⁹

... i wyjaśnij czemu decyzja o używaniu wątków powinna być dobrze uzasadniona?

Zadanie 8. Na podstawie §11.4.1 przedstaw model procesów i wątków systemu *Windows NT*, a następnie porównaj go z modelem uniksowym. Czy system *Linux* oferuje odpowiedniki **zadań** (ang. *jobs*), **pul wątków** (ang. *thread pool*), **włókien** (ang. *fiber*) i **planisty trybu użytkownika** (ang. *User-Mode Scheduling*)? Co to znaczy, że włókna są planowane kooperacyjnie (ang. *cooperative scheduling*)?

⁷<https://arstechnica.com/gaming/2006/11/valve-multicore/>

⁸<http://www.catb.org/esr/writings/taoup/html/ch07s03.html#id2923889>

⁹<https://web.stanford.edu/~ouster/cgi-bin/papers/threads.pdf>