

Systemy komputerowe

Lista zadań nr 9

Na zajęcia 25 kwietnia – 6 maja 2019

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Tanenbaum (wydanie czwarte): 1.1, 1.2, 1.4, 1.5
- Stallings (wydanie dziewiąte): 2.1 – 2.2, 13.3, 13.4

UWAGA! W trakcie prezentacji należy być gotowym do zdefiniowania pojęć oznaczonych **wytłuszczoną** czcionką.

Zadanie 1. Wyjaśnij różnice między **powłoką** (ang. *shell*), **systemem operacyjnym** i **jądrem systemu operacyjnego** (ang. *kernel*). W tym celu dobierz kilka przykładów powszechnie wykorzystywanego oprogramowania. Jakie są główne zadania systemu operacyjnego z punktu widzenia programisty?

Zadanie 2. Na podstawie **dokumentacji**¹ wymień składowe **pakietu** deb ze szczególnym uwzględnieniem zawartości pliku `control`. Porównaj zarządzanie zainstalowanym oprogramowaniem z użyciem pakietów i instalatorów znanych z systemów nieuniknowych. Weź pod uwagę proces pobierania, weryfikacji, instalacji, konfiguracji i odinstalowania oprogramowania.

Zadanie 3. Czym jest **zadanie** w **systemach wsadowych**? Jaką rolę pełni **monitor**? Na czym polega **planowanie zadań**? Zapoznaj się z rozdziałem „System Supervisor” dokumentu **IBM 7090/7094 IBSYS Operating System**². Wyjaśnij znaczenie poleceń **języka kontroli zadań** (ang. *Job Control Language*) użytych na rysunku 3 na stronie 13. Do jakich zastosowań używa się dziś systemów wsadowych?

Wskazówka: Bardzo popularnym systemem realizującym szeregowanie zadań wsadowych jest **SLURM**³.

Zadanie 4. Jaka była motywacja do wprowadzenia **wieloprogramowych** systemów wsadowych? W jaki sposób wieloprogramowe systemy wsadowe wyewoluowały w systemy z **podziałem czasu** (ang. *time-sharing*)? Podaj przykład systemu **interaktywnego**, który nie jest wieloprogramowy.

Zadanie 5. Bardzo ważną zasadą przy projektowaniu oprogramowania, w tym systemów operacyjnych, jest rozdzielenie **mechanizmu** od **polityki**. Wyjaśnij te pojęcia odnosząc się do powszechnie występujących rozwiązań, np. otwieranie drzwi klasycznym kluczem versus kartą magnetyczną.

Zadanie 6. Wymień mechanizmy sprzętowe niezbędne do implementacji **wywłaszczenia** (ang. *preemption*). Jak użyć **algorytmu rotacyjnego** (ang. *round-robin*) do implementacji wielozadaniowości z wywłaszczeniem? Jakie zadania pełni **planista** (ang. *scheduler*) i **dyspozytor** (ang. *dispatcher*)? Który z nich realizuje politykę, a który mechanizm?

Zadanie 7. Zapoznaj się z punktami podrozdziału „**The Elements of Operating-System Style**”⁴ książki „The Art of Unix Programming”. Opisz jak zaprojektować system sprzecznie z zasadami filozofii uniksowej. Czemu system operacyjny powinien (a) umożliwiać szybkie tworzenie procesów i łatwą komunikację międzyprocesową (b) przechowywać dane w plikach tekstowych, a nie binarnych (c) udostępniać szereg narzędzi programistycznych (d) oferować bogaty wybór programów działających w linii poleceń?

¹http://tldp.org/HOWTO/html_single/Debian-Binary-Package-Building-HOWTO/

²http://bitsavers.org/pdf/ibm/7090/C28-6248-7_v13_IBSYS_Dec66.pdf

³<https://slurm.schedmd.com/SC17/SlurmOverviewSC17.pdf>

⁴<http://www.catb.org/~esr/writings/taoup/html/ch03s01.html>

Zadanie 8. Podaj główne cele projektowe systemów operacyjnych przeznaczonych dla systemów **czasu rzeczywistego** (ang. *real-time*), **sieci sensorów** (ang. *wireless sensor networks*) i **systemów wbudowanych** (ang. *embedded systems*). Rozważ następujące systemy: **FreeRTOS**⁵, TinyOS (§13.4) i μ CLinux (§13.3). Ich obszar zastosowań znacznie różni się od klasycznych SO – jak wpłynęło to na zakres oferowanych funkcji?
Wskazówka: Takie systemy często obsługują sprzęt z małą ilością pamięci RAM i brakiem jednostki MMU lub pamięci trwałej.

⁵<http://www.freertos.org>