

Systemy operacyjne 2016

Lista zadań nr 1

Na zajęcia 6 października 2016

Pewna trudność w zajęciach z „Systemów operacyjnych” polega na opanowaniu sprawnego wyszukiwania treści w podręcznikach do wykładu, zasobach Internetu, w podręczniku systemowym (poleceniem **man**¹), plikach nagłówkowych w katalogu **/usr/include**² (poleceniem **grep** lub **ack**) i źródłach jądra **FreeBSD**³. Posiłkowanie się hasłami z polskojęzycznej Wikipedii jest niewskazane ze względu na liczne błędy merytoryczne, które tam występują.

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Tanenbaum (wydanie czwarte): 1.1, 1.2, 1.5, 1.6
- Stallings (wydanie siódme): 2.1 – 2.3

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

Zadanie 1. Na podstawie wybranych przez siebie przykładów wyjaśnij różnice między **powłoką**, **system operacyjnym** i **jądrem systemu operacyjnego**. Jakie są główne zadania systemu operacyjnego z punktu widzenia programisty?

Zadanie 2. Czym jest **dystrybucja** systemu operacyjnego? Przedstaw główne zadania **systemu pakietów** deb znanego z dystrybucji wywodzących się od Debian-a. Porównaj zarządzanie zainstalowanym oprogramowaniem z użyciem pakietów i instalatorów znanych z systemów nieunixowych.

Zadanie 3. 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 4. Wymień podstawowe funkcjonalności jądra systemu operacyjnego. Dla każdego z podanych **wywołań systemowych** podaj kilka warunków, przy których mogą się one zakończyć niepowodzeniem: **fork**, **kill**, **mount**, **unlink**, **read**. W jaki sposób system operacyjny komunikuje programiście przyczynę niepowodzenia?

Zadanie 5. Czym jest **zadanie** w **systemach wsadowych**? Jaką rolę pełni **monitor**? Na czym polega **planowanie zadań**? Wymień najważniejsze polecenia wybranego **języka kontroli zadań** (ang. *Job Control Language*) i wyjaśnij do czego są służyć. Czy w dzisiejszych czasach używa się systemów wsadowych. Jeśli tak, to do jakich zastosowań?

Zadanie 6. Jaka była motywacja do wprowadzenia **wieloprogramowych** systemów wsadowych? W jaki sposób wieloprogramowe systemy wsadowe ewoluowały w systemy z **podziałem czasu**? Czy **interaktywne** systemy operacyjne muszą być wieloprogramowe? Jeśli nie, to podaj przykład.

Zadanie 7. Podaj główne motywacje projektantów systemów operacyjnych do wprowadzenia **procesów** i **wątków**? Wymień główne różnice między nimi – rozważ współdzielone **zasoby**.

¹<https://www.freebsd.org/cgi/man.cgi>

²<http://bxx.su/FreeBSD/include/>

³<http://bxx.su/FreeBSD/sys/>

Zadanie 8. Wymień mechanizmy sprzętowe niezbędne do implementacji **wywłaszczenia**. Wyjaśnij jak użyć **algorytmu rotacyjnego** (ang. *round-robin*) do implementacji wielozadaniowości z wywłaszczeniem. Za co odpowiada **planista** (ang. *scheduler*), a za co **dyspozytor** (ang. *dispatcher*)?