

Systemy operacyjne

Ćwiczenia 1

Należy się przygotować do zajęć czytając następujące rozdziały książek:

- Stallings: 1.1 – 1.4, 2.1
- Tanenbaum: 1.1, 1.2, 1.4 – 1.6
- Silberschatz: 1

Pytania praktyczne wymagają znajomości systemu uniksowego (np. Linux, BSD, MacOS X). Można posługiwać się Wikipedią, ale polskojęzyczne hasła często zawierają błędy merytoryczne. Anglojęzyczne są w lepszym stanie, ale nie zastąpią czytania książek !

Zadanie 1

Zdefiniuj pojęcia: powłoka, system operacyjny i jądro systemu operacyjnego. Jakie są między nimi różnice? Jakie są główne zadania systemu operacyjnego?

Zadanie 2

Wymień podstawowe funkcjonalności jądra systemu operacyjnego. W tym celu zdefiniuj pojęcie wywołania systemowego (ang. *system call*) i powiedz do czego służy. Na podstawie SO implementujących interfejs POSIX wymień kilka grup wywołań systemowych.

Zadanie 3

Na podstawie formatu ELF (ang. *Executable and Linkable Format*) wyjaśnij z czego składa się plik wykonywalny. W tym celu wykorzystaj narzędzia `readelf` i `objdump` z pakietu *GNU Binutils*. Wy tłumacz do czego służą trzy podstawowe sekcje programu: `text`, `data` i `bss`.

Zadanie 4

Bardzo ważną zasadą przy projektowaniu systemów operacyjnych jest rozdzielenie *mechanizmu* od *polityki*. Wyjaśnij te pojęcia odnosząc się do rzeczywistości – np. fizyczny klucz do otwierania drzwi vs. karta magnetyczna.

Zadanie 5

W standardowej bibliotece języka C znajdują się funkcje `setjmp` i `longjmp`. Służą one do przeprowadzania nielokalnych skoków – wyjaśnij zasadę ich działania. Jakie dane muszą być zachowane w kontekście używanym przez te funkcje i dlaczego? Skorzystaj z podręcznika systemowego i pliku nagłówkowego `<setjmp.h>`.

Zadanie 6

Większość procesorów oferuje co najmniej dwa tryby pracy – użytkownika i nadzorcy – dlaczego? Znajdź informacje nt. instrukcji uprzywilejowanych¹ – wymień kilka aspektów działania procesora, które kontrolują. Czy jesteś w stanie uzasadnić szczególną rolę tych instrukcji?

¹ Najlepiej w dokumentacji zestawu instrukcji jakiegoś procesora (np. x86-64).

Zadanie 7

Wyjaśnij pojęcie przerwania. Opisz trzy klasy przerwania, które rozróżniamy pod względem źródła pochodzenia (okoliczności wygenerowania). Dla każdej klasy podaj konkretne przykłady zdarzeń. Jaka jest różnica pomiędzy przerwaniem sprzętowym, a programowym? Czy pułapki mogą być świadomie generowane przez program użytkownika – jeśli tak, to w jakim celu?

Zadanie 8

Opisz dokładnie mechanizm obsługi przerwania z uwzględnieniem informacji odkładanych na stos (zakładamy architekturę x86). Uwzględnij sytuacje, gdzie podczas obsługi jednego przerwania, pojawiają się inne. Czemu procesory przy obsłudze przerwania przechodzą w tryb nadzorcy i używają osobnego stosu? W jaki sposób procesor wie, którą z procedur obsługi przerwania zawołać przy danym zdarzeniu?