

Systemy operacyjne

Lista zadań nr 0

Na zajęcia 8–9 października 2019

Celem zadań z tej listy jest sprawdzenie czy student posiada wystarczającą wiedzę z przedmiotu „Architektury systemów komputerowych” wymaganą do studiowania „Systemów operacyjnych”.

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

Wskazówka: Prowadzący przedmiot zachęca studentów do pracy w grupie!

Zadanie 1. Opisz różnice między **przerwaniem sprzętowym** (ang. *hardware interrupt*), **wyjątkiem procesora** (ang. *exception*) i **pułapką** (ang. *trap*). Dla każdego z nich podaj co najmniej trzy przykłady zdarzeń, które je wyzwalają. W jakim scenariuszu wyjątek procesora nie oznacza błędu czasu wykonania programu? Kiedy pułapka jest generowana w wyniku prawidłowej pracy programu?

Zadanie 2. Opisz mechanizm **obsługi przerw** bazujący na **wektorze przerw** (ang. *interrupt vector table*). Co robi procesor przed pobraniem pierwszej instrukcji **procedury obsługi przerwania** (ang. *interrupt handler*) i po natrafieniu na instrukcję powrotu z przerwania? Czemu procedura obsługi przerwania powinna być wykonywana w **trybie nadzorcy** i używać odrębnego stosu?

Zadanie 3. Język C definiuje pewne słowa kluczowe wpływające na sposób dostępu do pamięci. Jakie jest znaczenie słowa kluczowego «volatile»? Wymień co najmniej dwa scenariusze, w których użycie wskaźników do ulotnej zawartości pamięci jest niezbędne dla poprawności działania programu.

UWAGA: Słowo kluczowe «volatile» nie wyłącza optymalizacji!

Zadanie 4. Bazując na formacie ELF (ang. *Executable and Linkable Format*) opisz składowe pliku wykonywalnego. Czym różni się **sekcja** od **segmentu**? Co opisują **nagłówki programu**? Skąd system operacyjny wie, pod jakim adresem ma umieścić segmenty programu i gdzie położona jest pierwsza instrukcja programu?

Wskazówka: Skorzystaj z narzędzia «readelf».

Zadanie 5. Zasada **lokalności odwołań** mówi o tym, że programy wykazują pewne regularności w dostępie do pamięci. Zdefiniuj pojęcia **lokalności czasowej** i **przestrzennej**, a następnie wytłumacz je używając pseudokodu podanego na tablicy. Opisz jak stosuje się tę zasadę w projektowaniu systemów komputerowych na przykładzie pamięci operacyjnej, dysku twardego i sieci Web.

Zadanie 6. Zapoznaj się z rozdziałami 3.4 i A.2 dokumentu [System V Application Binary Interface AMD64 Architecture Processor Supplement¹](#) i odpowiedz na następujące pytania:

- W jaki sposób jądro musi przygotować **przestrzeń adresową** procesu? Co musi znajdować się na stosie w momencie wywołania procedury «_start»? Do czego służy auxiliary vector? Można go wyświetlić wydając przykładowe polecenie «LD_SHOW_AUXV=1 /bin/true».
- W jaki sposób wywołać funkcję jądra? W których rejestrach należy umieścić argumenty? Gdzie można spodziewać się wyników i jak jądro sygnalizuje niepowodzenie **wywołania systemowego**?

Zadanie 7. Przypomnij jak wygląda mechanizm **tłumaczenia adresów** bazujący na dwupoziomowej tablicy stron w procesorze z rodziny IA-32. Przedstaw algorytm obliczania **adresu fizycznego** na podstawie **adresu wirtualnego** z uwzględnieniem uprawnień dostępu. Jaką rolę w procesie tłumaczenia odgrywa **pamięć TLB**?

Zadanie 8. Wyjaśnij znaczenie bitów: P, R/W, U/S, PWT, PCD, A, D, G; we **wpisie tablicy stron** (ang. *page table entry*) procesora z rodziny IA-32. W jakim celu stosuje je jądro systemu operacyjnego? Które z nich może modyfikować procesor i dlaczego?

Wskazówka: Zapoznaj się z rozdziałem czwartym dokumentu „Intel® 64 and IA-32 Architectures Software Developer’s Manual – Volume 3: System Programming Guide”.

¹https://www.uclibc.org/docs/psABI-x86_64.pdf