

Systemy operacyjne

Ćwiczenia 1

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- Stallings: 1.4 – 1.9, 1A, 7A, 8.1
- Harris & Harris: 8.3, 8.4

Dużo technicznych informacji znajduje się na [OSWiki](#).

Od tych ćwiczeń będziemy zakładać, że jesteście właściwie przygotowani z przedmiotu “Architektury Systemów Komputerowych”. W szczególności będą wam znane zagadnienia związane z pamięcią podręczną i wirtualną, przerwaniami, trybami pracy procesora i urządzeniami wejścia-wyjścia.

Uwaga! W trakcie prezentacji należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **pogrubioną czcionką**.

Zadanie 1

Opisz dokładnie **mechanizm obsługi przerwań** z uwzględnieniem informacji odkładanych na stos (dla architektury $\times 86$). Uwzględnij, że podczas obsługi danego przerwania mogą pojawić się inne. Czemu procesory przy obsłudze przerwań przechodzą w tryb nadzorcy i używają osobnego stosu? W jaki sposób procesor wie, którą z procedur obsługi przerwania zawołać przy danym zdarzeniu?

Zadanie 2

Wymień przynajmniej cztery (istotnie różne) obszary, które powinny być wyspecyfikowane przez **ABI** (ang. *Application Binary Interface*)? Przytocz kilka konwencji na podstawie dokumentu: [System V Application Binary Interface: AMD64 Architecture Processor Supplement](#). Dlaczego ustandaryzowane ABI jest niezbędne do implementacji systemu operacyjnego?

Zadanie 3

Kompilator języka C generuje pliki w assemblerze¹ dla konkretnej architektury. Oprócz instrukcji wraz z **operandami**, assembler pozwala na używanie **dyrektyw** i deklarowanie **symboli**. W jaki sposób udostępnić funkcję dla innych **modułów**? **Konsolidator** otrzymuje od użytkownika listę modułów i bibliotek celem utworzenia pliku wykonywalnego – na czym polega **wiązanie adresów** i **relokacja**?

Zadanie 4

Zasada lokalności odwołań mówi o tym, że programy wykazują pewne wzorce dostępu do komórek pamięci. Zdefiniuj pojęcia **lokalności czasowej** i **przestrzennej**, a następnie wytłumacz je na podstawie pseudokodu (podanego na tablicy). W jaki sposób wykorzystuje się tę zasadę w systemach komputerowych (na poziomie pamięci operacyjnej, dysków twardych i sieci Internet)?

Zadanie 5

Przypomnij zasadę działania pamięci podręcznej z **mapowaniem bezpośrednim**. Do czego służą dodatkowe dane skojarzone z **wierszem pamięci podręcznej** – znacznik i bity pomocniczne? W jaki sposób procesor adresuje pamięć podręczną? Opisz znaczenie bitów *dirty* oraz *valid* – pokaż scenariusz, który uzasadnia ich istnienie. Wyjaśnij dwie **polityki zapisu**: *write-back* i *write-through*.

¹ Można je podejrzeć dodając odpowiednią flagę kompilatora np.: `gcc -O2 -S -o test.s test.c`

Zadanie 6

Pamięć podręczna może używać mapowania bezpośredniego lub **skojarzeniowego**. Cemu w praktyce korzysta się z mapowania **sekcyjno-skojarzeniowego**? Procesor posiada cache sekcyjno-skojarzeniowy o 2^N wierszach, gdzie sekcja składa się z czterech wierszy. Jak realizowane są operacje odczytu i zapisu do pamięci – rozważ sytuację **trafienia**, **chybienia** i **konfliktu**. Opisz działanie **polityki zastępowania**: *LRU* i *random*.

Zadanie 7

Bez sprzętowego wsparcia dla zarządzania pamięcią, **adresy efektywne** są interpretowane jako **adresy fizyczne**. Po uruchomieniu systemu operacyjnego zarówno jądro SO jak i programy generują **adresy wirtualne**, które **tłumaczone** są do adresów fizycznych przez **jednostkę zarządzania pamięcią** (ang. *MMU*). Zastanów się jakie możliwości daje **sprzętowe tłumaczenie adresów** w porównaniu z architekturami, które nie dysponują takim rozwiązaniem?

Zadanie 8

Przypomnij jak wygląda **tłumaczenie adresów** w architekturze x86 – ograniczmy się do **stronicowania**. Na podstawie danego adresu efektywnego pokaż jak przeglądana jest **wielopoziomowa tablica stron** celem odnalezienia obliczenia adresu fizycznego. Wyjaśnij znaczenie bitów pomocniczych we **wpisie tablicy stron**. Jaką rolę w procesie translacji odgrywa **bufor TLB**?

Zadanie 9

Opisz mechanizm **bezpośredniego dostępu do pamięci** (ang. *Direct Memory Access*). Narysuj diagram architektury prostego komputera z uwzględnieniem DMA. Odpowiedz na pytania:

- Jakie dane są potrzebne **kontrolerowi DMA** do rozpoczęcia transakcji **transferu**?
- W jaki sposób procesor jest informowany o zakończeniu transferu?
- Załóżmy że, procesor może wykonywać instrukcje podczas trwania transmisji DMA. Czy transmisja może wpływać na szybkość wykonywania programów?