

# Systemy operacyjne

## Ćwiczenia 2

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, pamięcią 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

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 dla implementacji systemu operacyjnego?

### Zadanie 2

Kompilator języka C generuje pliki w asemblerze<sup>1</sup> dla konkretnej architektury sprzętowej. Oprócz instrukcji wraz z **operandami**, asembler 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 – opisz na czym polega **wiązanie adresów** i **relokacja**?

### Zadanie 3

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

### Zadanie 4

Przypomnij zasadę działania pamięci podręcznej z mapowaniem bezpośrednim. Do czego służą dodatkowe dane skojarzone z **linią pamięci podręcznej** – znacznik i bity pomocniczne? Opisz znaczenie bitów *dirty* oraz *valid* – pokaż scenariusz, który uzasadnia ich istnienie. Wyjaśnij dwie **polityki zapisu**: *write-back* i *write-through*; oraz dwie **polityki zastępowania**: *LRU* i *random*.

---

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

### Zadanie 5

Pamięć podręczna może używać mapowania **bezpośredniego** lub **skojarzeniowego**. Czemu w praktyce korzysta się z mapowania **sekcyjno-skojarzeniowego**? W jaki sposób procesor adresuje pamięć podręczną? Dla powyższych organizacji cache podaj jak procesor korzysta z pamięci podręcznej w trakcie odczytu i zapisu – rozważ sytuację **trafienia, chybienia i konfliktu**.

### Zadanie 6

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*). Jakie możliwości daje takie rozwiązanie w porównaniu z architekturami bez sprzętowego tłumaczenia adresów?

### Zadanie 7

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 8

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 odczytu / zapisu ?
- 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?