

Wykład jest elementarnym wprowadzeniem do budowy i zasad działania komputerów z punktu widzenia programisty. Po zaliczeniu przedmiotu studenci powinni rozumieć jaki wpływ ma architektura komputera na wydajność aplikacji, które przyjdzie im implementować w praktyce zawodowej.

Program:

Architektura programowa [16h]

- Organizacja komputera (składowe systemu, ewolucja, RISC vs. CISC, ograniczenia).
- Reprezentacja danych w pamięci komputera (integer, float, string, audio, image).
- Reprezentacja programów w pamięci komputera (MIPS ISA, debugowanie, deasemblacja).
- Programowanie strukturalne (warunki, pętle, stos, procedury, wskaźniki, tablice, struktury, unie).
- Proces kompilacji i konsolidacji (symbole, sekcje, relokacja, ABI, pliki ELF, układ pamięci).
- Tryby pracy procesora (model użytkownika i systemowy, zasoby i instrukcje uprzywilejowane).
- Przerwania i wyjątki (obsługa, wektor przerwań, priorytety, NMI, tryb krokowy).
- Dostęp do urządzeń wejścia-wyjścia (MMIO, odpytywanie, przerwania, DMA).

Pamięć [6h]

- Organizacja pamięci operacyjnej (czasu dostępu, protokół DDR).
- Dyski magnetyczne i półprzewodnikowe (multi-zone recording, buforowanie, odświeżanie bloków).
- Pamięć podręczna (organizacja, polityka wymiany i zapisu).
- Tłumaczenie adresów (stronicowanie, wielopoziomowe tablice stron, TLB).

Architektura procesora [4h]

- Elementarz mikroarchitektury (układy kombinacyjne i sekwencyjne, przerzutnik, multiplexer, ALU).
- Procesor jednocyfrowy.
- Przetwarzanie potokowe (etapy przetwarzania instrukcji MIPS, hazardy danych i kontroli).

Przetwarzanie równoległe [4h]

- Superskalarność (architektura Out-of-Order, spekulacja, przemianowywanie rejestrów, SMT).
- Przewidywanie skoków (BHT, BTAC, przewidywanie lokalne i globalne).

- Systemy SMP (instrukcje atomowe, komunikacja międzyprocesorowa, cache snooping).
- Przetwarzanie wektorowe (SIMD, procesory wektorowe, GPGPU).

Literatura:

- Computer Organization and Design: The Hardware/Software Interface; piąte wydanie; Hennesy, Patterson
- Computer Systems: A Programmer's Perspective; trzecie wydanie; Randal E. Bryant, David R. O'Hallaron

Zajęcia:

Większość zajęć do przedmiotu będzie prowadzona w postaci ćwiczeń z deklaracjami. Pozostałe to pracownie programistyczne (programowanie w asemblerze MIPS) i projekty programistyczne (prosty asembler lub symulator procesora MIPS, symulator pamięci podręcznej i predyktora skoków).