Pregled računarskog sistema

Operativni sistemi 1

Institut za matematiku i informatiku
Prirodno-matematički fakultet, Kragujevac

doc. dr Miloš Ivanović
prof. dr Dragan Stojanović

Oktobar 2026. god.

O čemu će biti reči?

1. O predmetu
2. Struktura
3. CPU
4. Prekidi
5. Memorija
   - Keš
6. U/I
7. Multiprocesor

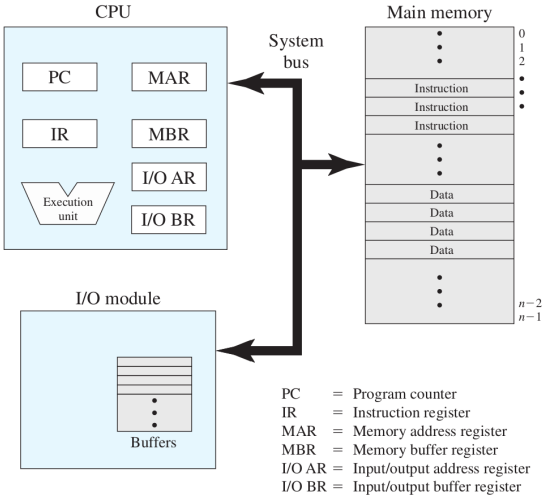
O predmetu

- Nastavnik: **Dragan Stojanović**
- Asistenti: **Filip Andrić, Tomislav Manojlović**
- Model 50+50 poena
- Završni ispit se sastoji iz 10 pitanja i polaže se pismeno.

Literatura:

1. **William Stallings**:, *Operating Systems: Internals and Design Principles, 9th edition, Pearson Education 2017* (Operativni sistemi: Principi unutrasnje organizacije i dizajna), prevod 9. izdanja, CET, 2019.
2. **B. Đorđević, D. Pleskonjić, N. Maček**: "Operativni sistemi: teorija, praksa i rešeni zadaci", Mikro knjiga, Beograd, 2005.
3. **B. Đorđević, D. Pleskonjić, N. Maček**: "Operativni sistemi: UNIX i Linux", Viša elektrotehnička škola, Beograd, 2004.

Delovi računara



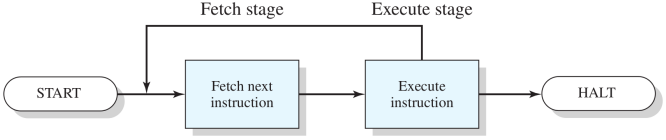
Registri mikroprocesora

1. Registri vidljivi i registri nevidljivi korisniku
2. Upravljački i statusni registri
3. Indeksni registar
4. Pokazivač segmenta
5. Pokazivač steka
6. Na većini procesora, pozivi procedura rezultuju automatskim stavljanjem na stek svih registara vidljivih korisniku
7. Program Counter (PC) i Instruction Register (IR)
8. PSW (Processor Status Word)

Instrukcijski ciklus

Instrukcijski ciklus

Instrukcijski ciklus se sastoji iz dva koraka, i to faze donošenja (*fetch*) i faze izvršavanja (*execute*).
Donešene instrukcije se učitavaju u IR, a PC se povećava tako da pokazuje na sledeću instrukciju.

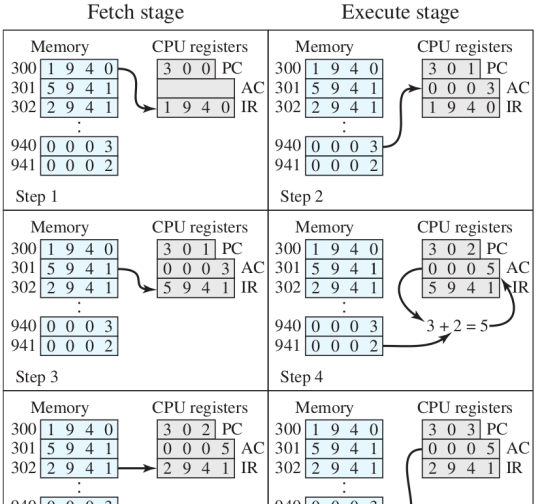


Instrukcijski ciklus

1. Transfer CPU-memorija i obratno
2. Transfer CPU-U/I - prenos podataka između CPU i U/I modula
3. Obrada podataka - aritmetičke i logičke operacije nad podacima
4. Upravljanje - instrukcije koje mogu da dovedu do promene redosleda izvršavanja

Izvršavanje programa na hipotetičkom računaru

1. Donošenje vrednosti sa adrese 940 u akumulator
2. Sabiranje vrednosti iz akumulatora sa podatkom na adresi 941
3. Prenos rezultata na adresu 941



Prekidi (Interrupts)

Primer

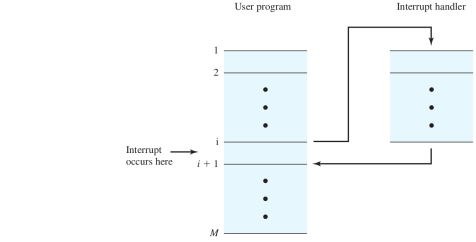
Osnovna namena prekida je da se poboljša iskorišćenost procesora. Preračunati razliku brzine između CPU koji radi na 3 *GHz* i diska koji radi na 7200 *rpm*.

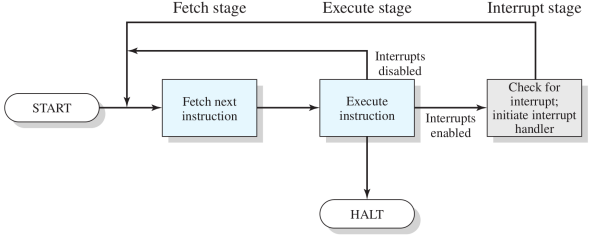
Klase prekida

1. **Program** - prekoračenje, deljenje nulom, segmentacija, **sistemski poziv**
2. **Tajmer** - Generiše ga tajmer unutar CPU. OS na taj način može redovno da izvršava neke instrukcije
3. **UI** - Generiše ga UI kontroler
4. **Otkaz hardvera** - npr. greška pariteta memorije

Prekidi (Interrupts)

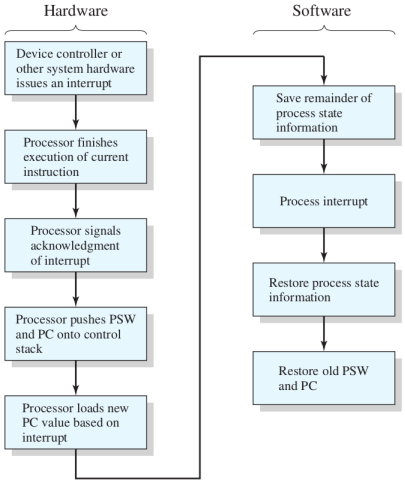
Obrada prekida i programski ciklus sa prekidom





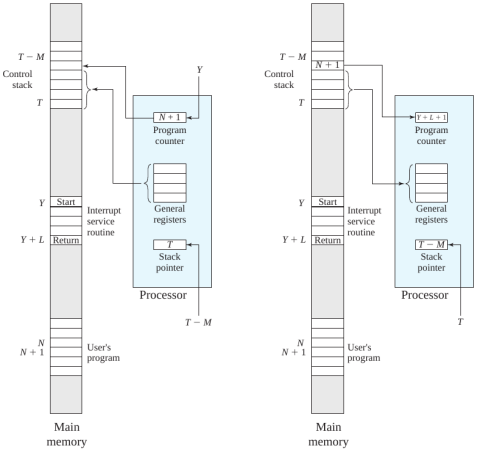
Prekidi (Interrupts)

Algoritam obrade prekida



Prekidi (Interrupts)

Šta se dešava u memoriji prilikom obrade prekida?



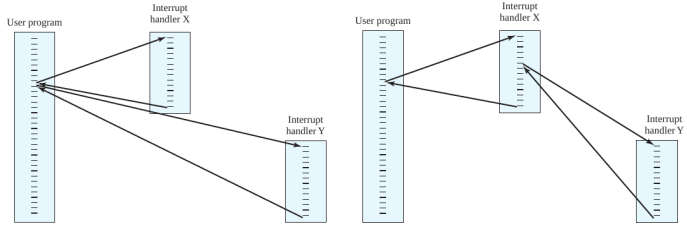
(a) Interrupt occurs after instruction at location N

(b) Return from interrupt

Višestruki prekidi

Vrste višestrukih prekida

Prekid može nastati i tokom izvršavanja neke prekidne rutine. Ovako nastali prekidi se mogu obraditi na **sekvencijlni** ili **ugnežđeni** način.



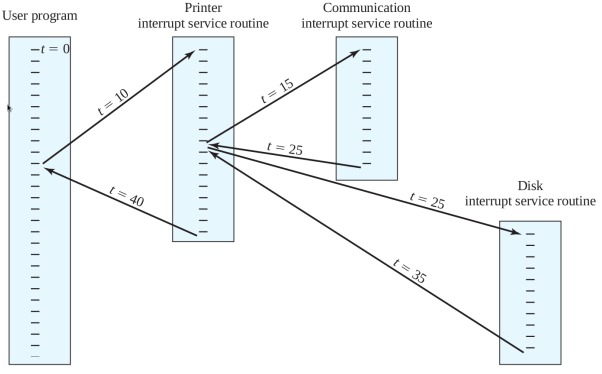
(a) Sequential interrupt processing

(b) Nested interrupt processing

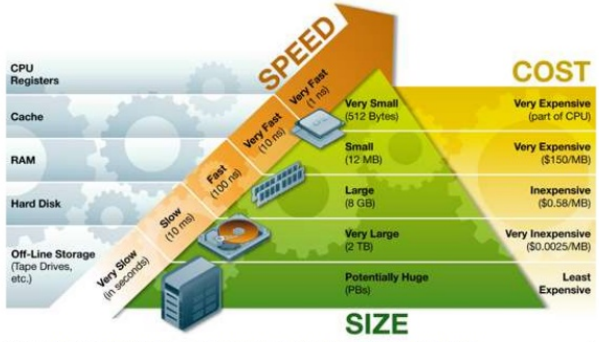
Organizacija prekida po prioritetu

Prioriteti prekida

Prekid višeg prioriteta može da izazove prekid rada prekida sa nižim prioritetom. Oznaka t na slici je vreme.



Hijerarhija memorija



Source: http://www.ts.avnet.com/uk/products_and_solutions/storage/hierarchy.html

Hijerarhija memorije

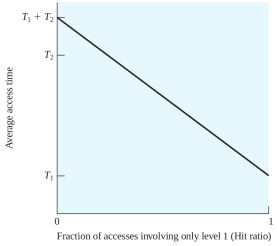
Odnosi koji uglavnom važe

1. Brže vreme pristupa - veća cena po bitu
2. Veći kapacitet - manji troškovi po bitu
3. Veći kapacitet - manja brzina pristupa

Spuštanje niz hijerarhiju

1. Smanjuje se cena po bitu
2. Povećava se kapacitet
3. Povećava se vreme pristupa
4. Smanjuje se učestanost pristupa memoriji

Primer keširanja memorije sa dva nivoa



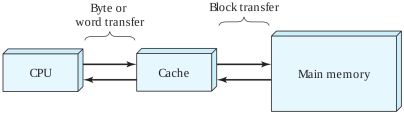
Primer

- **Nivo 1** - 100 bajtova sa vremenom pristupa 0.1 μs
- **Nivo 2** - 100000 bajtova sa vremenom pristupa 1 μs

Ako pretpostavimo da se 95% pristupa memoriji nalazi u kešu, prosečno vreme pristupa se može izraziti kao:

$$0.95 \cdot 0.1\mu s + 0.05 \cdot (0.1\mu s + 1\mu s) = 0.095 + 0.055 = 0.15\mu s$$

Kako funkcioniše keš

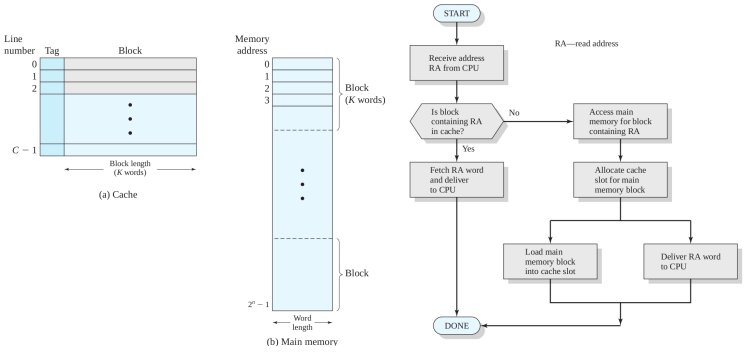


Implemenacija keša

- Glavna memorija - 2^n reči, gde svaka ima n-bitnu adresu. Sastoji se od **blokova** fiksne dužine K reči. Postoji $M = 2^n/K$ blokova.
- Keš memorija - C slotova koji imaju po K reči, $C << M$

Svaki slot sadrži **tag** (oznaku) koji ukazuje na to koji je trenutno blok smešten u njega. **Tag** se odnosi na sve adrese koje počinju tom sekvencom bitova.

Kako funkcioniše keš



Slika: Levo je šematski dato preslikavanje RAM-keš, a desno algoritam po kome procesor čita keš/RAM. Posebnu ulogu igra algoritam zamene.

.

U/I tehnike

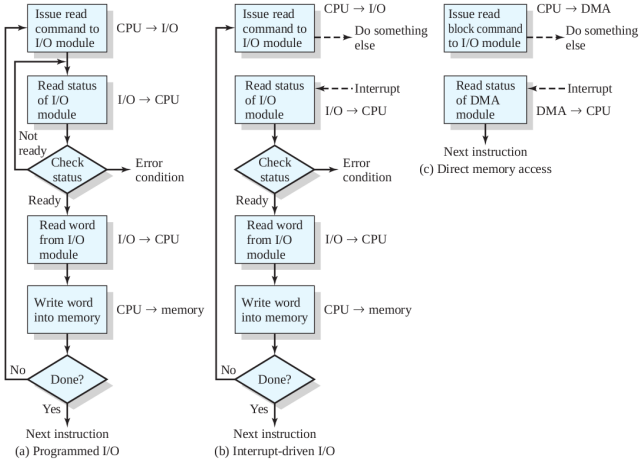
Tehnike koje se koriste za U/I operacije

1. Programirani U/I
2. U/I koji se upravlja prekidima (*interrupts*)
3. Direktan pristup memoriji (DMA)

Skup instrukcija U/I

1. **Upravljanje** - aktiviranje uređaja i davanje instrukcija
2. **Status** - testiranje stanja UI modula i njegovih periferala
3. **Prenos** - čitanje/upis podataka između registara CPU-a i eksternih uređaja

U/I tehnike


Issue read command to I/O module
CPU → I/O
Read status of I/O module
I/O → CPU
Not ready
Check status
Error condition
Ready
Read word from I/O module
I/O → CPU
Write word into memory
CPU → memory
No
Done?
Yes
Next instruction
(a) Programmed I/O
Issue read command to I/O module
CPU → I/O
Do something else
Read status of I/O module
Interrupt
I/O → CPU
Check status
Error condition
Ready
Read word from I/O module
I/O → CPU
Write word into memory
CPU → memory
No
Done?
Yes
Next instruction
(b) Interrupt-driven I/O
Issue read block command to I/O module
CPU → DMA
Do something else
Read status of DMA module
Interrupt
DMA → CPU
Next instruction
(c) Direct memory access

Multiprocesorski sistemi (SMP)

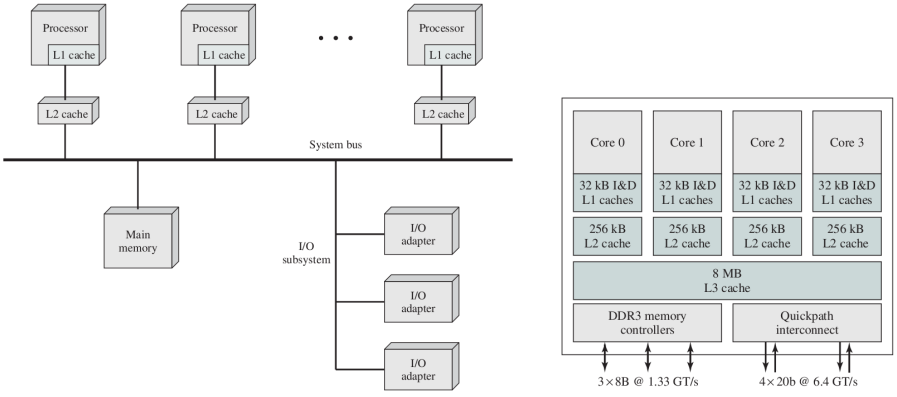
SMP organizacija

- U rač. sistemu se nalazi dva ili više procesora istih ili sličnih kapaciteta
- Svi procesori dele glavnu memoriju i imaju podjednako vreme pristupa
- Svi procesori dele pristup U/I uređajima kroz deljene ili posebne kanale
- OS je zadužen za interakciju između procesora

Prednosti SMP-a

1. Performanse
2. Dostupnost
3. Inkrementalni rast
4. Skalabilnost

Multiprocesorski sistemi (SMP)



Slika: *Levo*: Opšta SMP arhitektura, *Desno*: Intel Core i7 arhitektura

.