

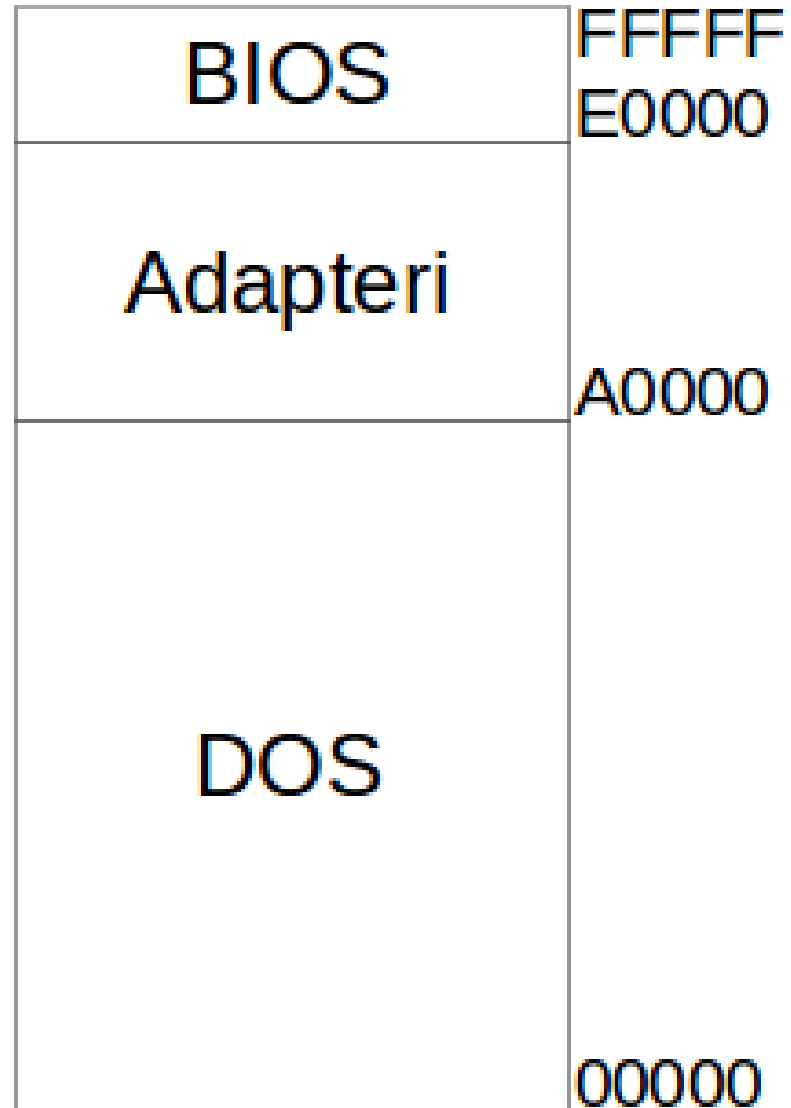
Virtuelna memorija

Literatura: Hennessy & Patterson, 2.4, B.4

Jednoprocesni sistem

- Ograda: sistem koji sinhrono reaguje na spoljašnje događaje nikad ne može da bude striktno jednoprocesni
- Proces vidi fizičku memoriju, kako je hardver mapira u adresni prostor procesora
 - \pm memorijski mapirani uređaji
- Sve memorijske adrese su fizičke adrese
- Ovo stvara poteškoće:
 - ako je potrebno pokretati pozadinske procese
 - ako memorijski raspored nije fiksni

Primer: MS-DOS



Problemi kod višeprocenog rada

- Nestabilnost adresnog prostora
 - apsolutna odredišta grananja ne mogu da se koriste u fiksnom obliku
 - reference na statički alocirane podatke takođe
 - rešava se dinamičkom relokacijom
- Problem stabilnosti sistema
 - proces može da pristupi osetljivim delovima memorije (npr. mapiranja za uređaje, prekidni vektori)
 - ovo može da izazove nestabilnost ili krah
- Problem bezbednosti/privatnosti
 - svaki proces može da vidi podatke svih drugih
 - podnošljivo kod jednokorisničkog sistema, nedopustivo kod višekorisničkog

Upravljanje memorijom

- Za rešavanje navedenih problema uvodi se koncept upravljanja memorijom
- Poseban podsistem na procesoru koji se zove jedinica za upravljanje memorijom (*Memory Management Unit*, MMU)
- Jedan od zadataka je da se memorija organizuje tako da svaki proces ima:
 - sliku raspoložive memorije kao da je jedini koji se izvršava na sistemu
 - memorijski prostor zaštićen od drugih procesa

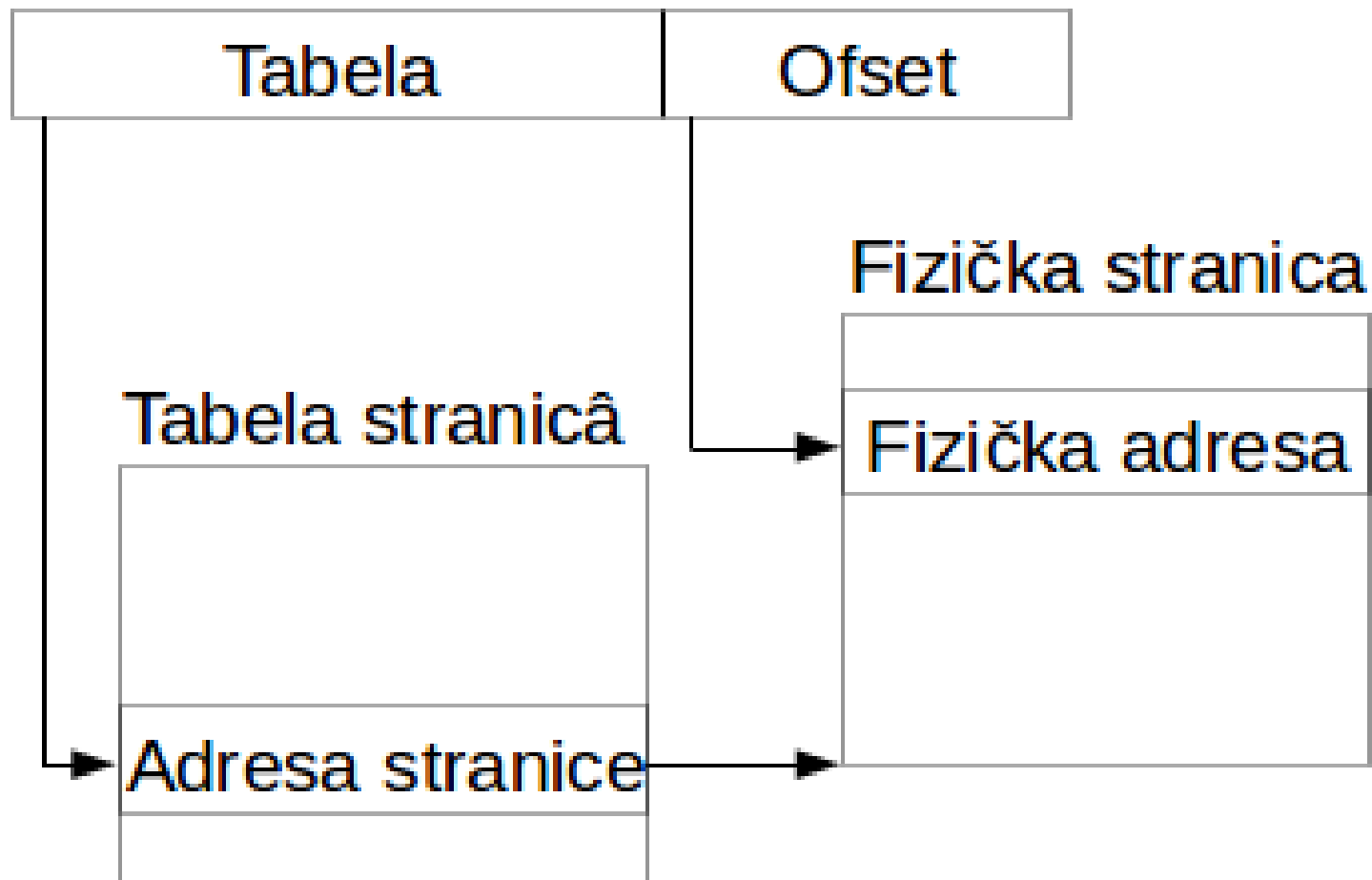
Virtuelna memorija

- Uvođenje MMU omogućava virtuelizaciju memorije
- Zaštita procesa je samo jedan (ali veliki) zadatak virtuelizacije
 - drugi je omogućavanje rada sa adresnim prostorom većim od raspoložive fizičke memorije
 - korišćenje sekundarne memorije (disk) kao podrške za memorijski sadržaj
 - mehanizam sličan keširanju
- Memorijske reference koje program koristi više se ne odnose direktno na fizičku memoriju
 - sve adrese su virtuelne adrese
 - mora postojati način za prevođenje iz virtuelnih u fizičke adrese

Principi prevođenja

- Memorija se deli na *stranice* (*pages*, 4 KB – 4 MB)
- Mora postojati memorijska struktura koja pokazuje na pojedinačne stranice
- Ova struktura se zove tabela stranica (*Page Directory*)
- Mogući su različiti pristupi prevođenju, u zavisnosti od veličine i broja stranica koje treba pokriti

Prosto prevodenje (1)



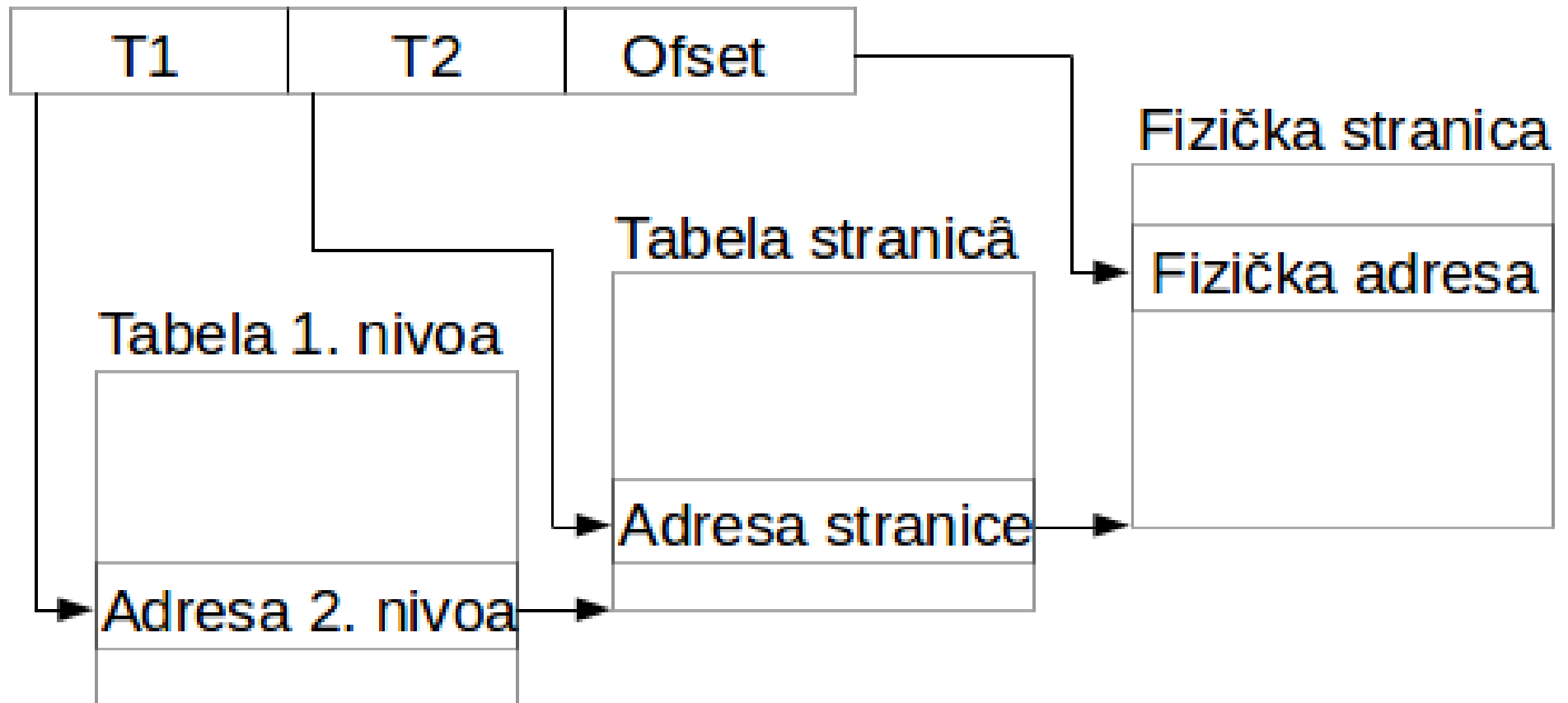
Prosto prevođenje (2)

- Veoma nalik na keširanje
 - pokazivač na tabelu stranica = tag
 - ofset u okviru stranice = ofset u okviru linije
- Realan primer: stranice od 4 MB
 - 22 bita za ofset, 10 bita za indeks stranice
 - 1024 elementa u tabeli

Prosto prevođenje (3)

- U opštem slučaju, velike stranice su neefikasne
 - veliki broj operacija zahteva poravnanje na veličinu stranice, što povećava utrošak memorije
- Za manje stranice, jedan nivo je takođe neefikasan
 - tabela stranica je velika
 - svakom procesu je potrebna posebna tabela
 - posledica: velika potrošnja memorije na tabele za upravljanje memorijom(!)

Više nivoa prevođenja (1)



Više nivoa prevođenja (2)

- Podelom adrese stranice na više delova omogućava se izbegavanje zauzimanja memorije za nealocirane delove u memorijskoj mapi
- Kod realne organizacije procesa čest je slučaj da su zauzeti disjunktne blokovi memorije na suprotnim krajevima adresnog prostora
- Ova organizacija omogućava i korišćenje većeg fizičkog adresnog prostora u odnosu na adresni kapacitet procesora

Više nivoa prevođenja (3)

- Tabele su hijerarhijske
- Proces konstruisanja prevođenja je nužno iterativan, ne može se paralelizovati
- Ako je MMU aktivan, prevođenje je potrebno prilikom svakog memorijskog pristupa
 - za dva nivoa, dva pristupa memoriji
 - tabele mogu imati do četiri nivoa
 - ako su tabele u L1d, bar četiri dodatna ciklusa po pristupu
- Potreban je način da se ovaj postupak ubrza

TLB keš

- Rešenje koje se primenjuje jeste da se kešira ukupan rezultat prevođenja: fizička adresa stranice
- Keš u kome se drže ovi podaci zove se TLB (= *Translation Lookaside Buffer*)
- Ovaj keš mora biti veoma brz da bi se izbeglo čekanje prilikom svakog memorijskog pristupa
- Obično realizovan kao potpuno asocijativni, sa malim brojem elemenata