Operacijski sustavi

Četvrta laboratorijska vježba

18. svibnja 2022.

Sadržaj

1 Motivacija

Cilj ove laboratorijske vježbe je proučiti rad sustava straničenja i izolacije adresnih prostora procesa u modernim računalima. U sklopu vježbe potrebno je ostvariti simulaciju rada više procesa na jednostavnom računalu koje koristi straničenje.

2 Zadatak

Vaš je zadatak ostvariti simulaciju rada više procesa u sustavu koji koristi mehanizam straničenja na zahtjev pomoću arhitekture prikazane na slici 1.

Simulirani sustav se sastoji od N procesa, diska, niza od M okvira i tablice straničenja za svaki simulirani proces. Vaš program kao ulazne parametre prima broj okvira M i broj procesa N te uz ponašanje navedeno u pseudokodu 1 mora sadržavati sljedeće strukture podataka:

- `disk[N]` - Simulirani disk koji služi za pohranu sadržaja stranica,
- `okvir[M]` - Simulirani radni spremnik od M okvira veličine 64 okteta,
- `tablica[N]` - Tablica prevođenja za svaki od N procesa.

Algoritam 1: Pseudokod simulacije

```
1  za i = 1 do N čini
2  |  stvori proces i;
3  |  inicijaliziraj tablicu straničenja procesa i;
4  kraj
5  t ← 0;
6  ponavljaj
7  |  za svaki proces p čini
8  |  |  x ← nasumična logička adresa;
9  |  |  nađi zapis tablice straničenja procesa p za adresu x;
10 |  |  ako adresa x nije prisutna onda
11 |  |  |  ispiši promašaj;
12 |  |  |  pronađi i dodijeli okvir;
13 |  |  |  učitaj sadržaj stranice s diska;
14 |  |  |  ažuriraj tablicu prevođenja procesa p;
15 |  |  kraj
16 |  |  ispiši adresu x, adresu njenog okvira te sadržaj zapisa u
   |  |   tablici prevođenja;
17 |  |  dohvati sadržaj adrese x;
18 |  |  inkrementiraj dohvaćeni sadržaj i zapiši ga na adresu x;
19 |  |  t ← t + 1;
20 |  |  spavaj;
21 |  kraj
```

Svaki zapis unutar tablice straničenja mora ostvarivati bit prisutnosti u šestom bitu zapisa.

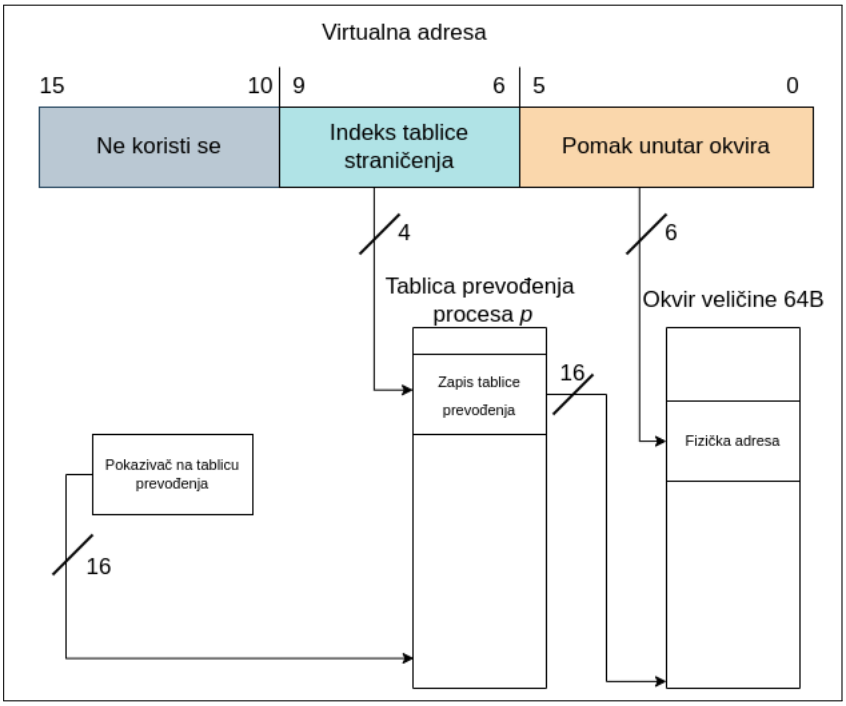
Prilikom pronalaženja slobodnog okvira za dodjeljivanje potrebno je koristiti *Least Recently Used (LRU)* strategiju zamjene stranica. Za implementaciju *LRU* strategije potrebno je koristiti varijablu `t` čija se vrijednost koristi kao sat. U strukturi zapisa tablice prevođenja (prikazan na slici 2) predviđeno je 5 bitova za pohranu *LRU* metapodataka za stranicu. Ako vrijednost tog polja u bilo kojoj stranici dođe na 31, potrebno je postaviti vrijednost globalne varijable `t` na 0, vrijednosti svih *LRU* metapodataka za sve zapise u svim tablicama straničenja na 0, te *LRU* metapodatak za trenutnu stranicu na 1. Prilikom zamjene odnosno izbacivanja stranica pretpostavite da je njihov sadržaj uvijek mijenjan te ga pohranite na disk.

Za potrebe generiranja nasumične logičke adrese preporučamo da generirate samo **parne** adrese kako bi izbjegli problematične rubne slučajeve sa čitanjem na krajevima okvira, što možete postići primjenom logičke opera-

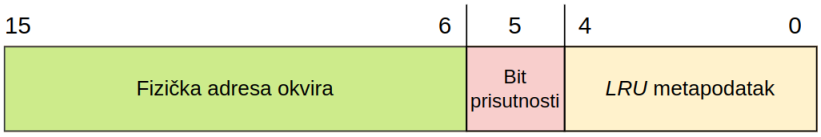
cije **I** sa vrijednosti `0x3FE` na generirani broj.

2.1 Kratki opis simulirane arhitekture

Simulirani procesor koristi 16-bitne logičke adrese s veličinom okvira/stranice od 64 okteta. Pretpostavite da se brojevi spremaju prema *little-endian* principu. Struktura virtualne adrese prikazana je na slici 1. Za potrebe prevođenja virtualnih adresa koristi se tablica straničenja s jednom razinom. Radi lakše simulacije prvih 6 bitova logičke adrese se **ne koristi**, čime je logički adresni prostor **ograničen na 1024 okteta**. Iduća 4 bita virtualne adrese koriste kao indeks u tablici straničenja, a zadnjih 6 bitova koriste se kao pomak unutar fizičkog okvira.



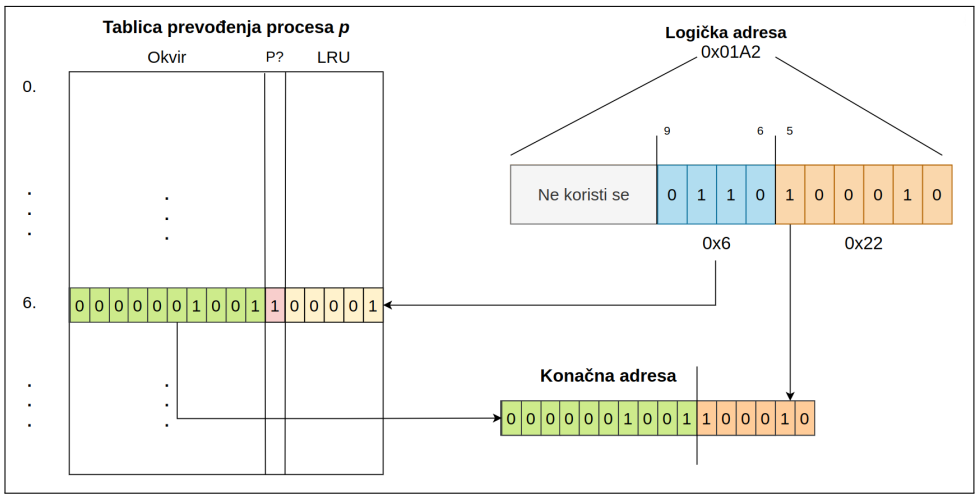
Slika 1: Dijagram sustava straničenja simuliranog računala.



Slika 2: Struktura zapisa u tablici prevođenja.

2.2 Primjer jednog koraka simulacije

Pretpostavimo da smo nasumično generirali adresu `0x1A2`, da je ta stranica prisutna te da je vrijednost varijable $\mathbf{t} = 3$. Kako bi pronašli odgovarajući zapis u tablici prevođenja, potrebno je razložiti adresu na polja specificirana na slici 1. Postupak je prikazan na slici 3. Vrijednost dobivenog indeksa je 6 te provjeravamo sedmi zapis u tablici (brojimo od nule). Razlaganjem vrijednosti sedmog zapisa uočavamo da je adresa okvira 0x9, da je bit prisutnosti postavljen na 1 te da je prethodna *LRU* vrijednost 1. Adresa podatka se potom formira iz adrese okvira i pomaka unutar okvira, a *LRU* podatak u zapisu tablice postavlja se 3 (trenutna vrijednost varijable **t** odnosno sata).



Slika 3: Primjer prevođenja adrese.

2.3 Primjer ispisa

Pretpostavimo da simuliramo dva procesa sa jednim dostupnim okvirom te da su u četiri iteracije simulacije oba procesa pristupala adresi `0x1FE`.

Primjer 1: Ispis simulacije.

```
$ ./lab4 2 1
---------------------------
proces: 0
        t: 0
        log. adresa: 0x01fe
        Promasaj!
                dodijeljen okvir 0x0000
        fiz. adresa: 0x003e
        zapis tablice: 0x0020
        sadrzaj adrese: 0

---------------------------
proces: 1
        t: 1
        log. adresa: 0x01fe
        Promasaj!
                Izbacujem stranicu 0x01c0 iz procesa 0
                lru izbacene stranice: 0x0000
                dodijeljen okvir 0x0000
        fiz. adresa: 0x003e
        zapis tablice: 0x0021
        sadrzaj adrese: 0

---------------------------
proces: 0
        t: 2
        log. adresa: 0x01fe
        Promasaj!
                Izbacujem stranicu 0x01c0 iz procesa 1
                lru izbacene stranice: 0x0001
                dodijeljen okvir 0x0000
        fiz. adresa: 0x003e
        zapis tablice: 0x0022
        sadrzaj adrese: 1
---------------------------
proces: 1
        t: 3
        log. adresa: 0x01fe
        Promasaj!
                Izbacujem stranicu 0x01c0 iz procesa 0
                lru izbacene stranice: 0x0002
                dodijeljen okvir 0x0000
        fiz. adresa: 0x003e
        zapis tablice: 0x0023
        sadrzaj adrese: 1
^C
```