

## Virtuelna memorija

Virtuelna memorija je tehnika koja dozvoljava izvršavanje procesa čiji delovi mogu biti smešteni na sekundarnim memorijama, tj diskovima. Virtuelna memorija formira apstrakciju u vidu logičke memorije, koju čine radna memorija i sekundarna memorija i razdvaja korisničku logičku memoriju od fizičke. Količina raspoložive fizičke memorije više ne ograničava program, pa programeri mogu da pišu programe bez korišćenja tehnike preklapanja (*overlay*). Pri tome, virtuelna memorija omogućava izvršavanje programa većih od same fizičke memorije. Koncept virtuelne memorije omogućava smeštanje osetno većeg broja procesa u memoriju (konkretno delova procesa), čime se povećavaju iskorišćenje i propusna moć procesora, a bez povećanja vremena odziva (*response time*) i vremena izvršavanja (*turnaround time*). Virtuelna memorija omogućava deljenje datoteka i memorije između različitih procesa na isti način kao i kod deljenja stranica, što omogućava uštedu memorije i poboljšanje performansi.

### Učitavanje stranica po potrebi

Virtuelna memorija se najčešće realizuje tehnikom učitavanja stranica prema potrebi (*demand paging*, DP), a mogu se upotrebljavati i tehnike učitavanja segmenata prema potrebi (*demand segmentation*) – koja je dosta složenija – kao i kombinovana tehnika.

Sistem sa učitavanjem stranica prema potrebi veoma je sličan straničenju sa razmenjivanjem (*swap*). I memorija i prostor na disku koji se koristi za razmenjivanje izdeljeni su na stranice. Procesi su smešteni na disku u swap prostoru i prebacuju se u memoriju kada dođu na red za dodelu procesora. Prvi sistemi koji su koristili swap tehniku, prebacivali su isključivo cele procese iz swap prostora u fizičku memoriju i obratno. Za razliku od njih, DP sistem koristi specifičnu tehniku za razmenjivane: u fizilku memoriju se ne prebacuje ceo proces, nego samo stranice koje se trenutno traže i to najčešće samo ona koja je neophodna. Na taj način se izbegava nepotrebno čitanje sa diska i smanjuje potrebna količina fizičke memorije. DP može prebacivati i više stranica u memoriju na osnovu pretpostavke o stranicama koje će biti potrebne procesoru. DP je tzv "lenji razmenjivač" (*lazy swapper*), jer nikada ne prebacuje stranicu u memoriju, osim ako se stranice ne zahtevaju. Proses koji obavlja razmenu naziva se *pager*, a sistem koji koriste DP – *paging* sistemi.

### Neophodan hardver

Za realizaciju DP tehnike neophodni su hardver za straničenje i hardver za ramenjivanje. Sledeće dve komponente su apsolutno obavezne:

- Tabela stranica
- Sekundarna memorija

Eksplicitno se zahteva prisustvo bita validnosti u tabeli stranica. Pomoću bita validnosti, koji se čuva u tabeli stranica, može se opisati trenutni položaj stranica. Vrednost bita *v* (*valid*), ukazuje na to da se logička stranica nalazi u memoriji. Vrednost bita *i* (*invalid*), ukazuje na to da se stranica ne nalazi u memoriji, već na disku, u swap prostoru. Vrednost *i* takođe može da označavati da stranica ne pripada adresnom prostoru diska.

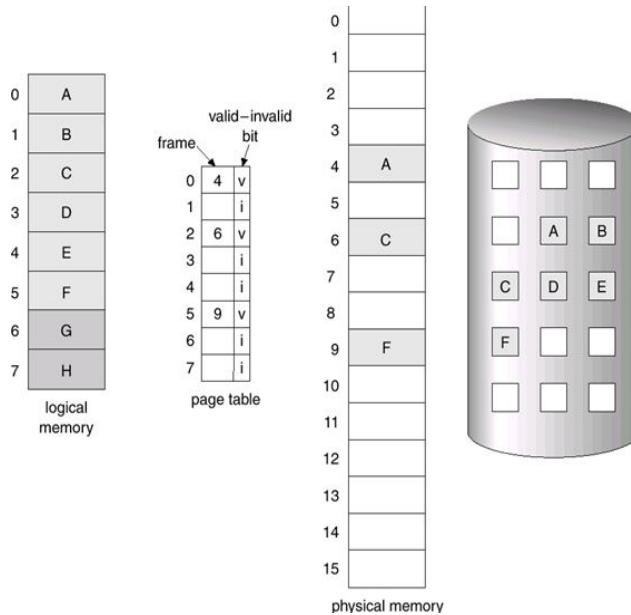
Sekundarna memorija služi za smeštaj svih stranica koje nisu u memoriji. Po pravilu se koristi disk.

### Prebacivanje stranica sa diska u fizičku memoriju

DP je ilustrovan na slici, na kojoj je prikazan izgled logičkog adresnog prostora hipotetičkog procesa, odgovarajuće fizičke memorije i swap prostora.

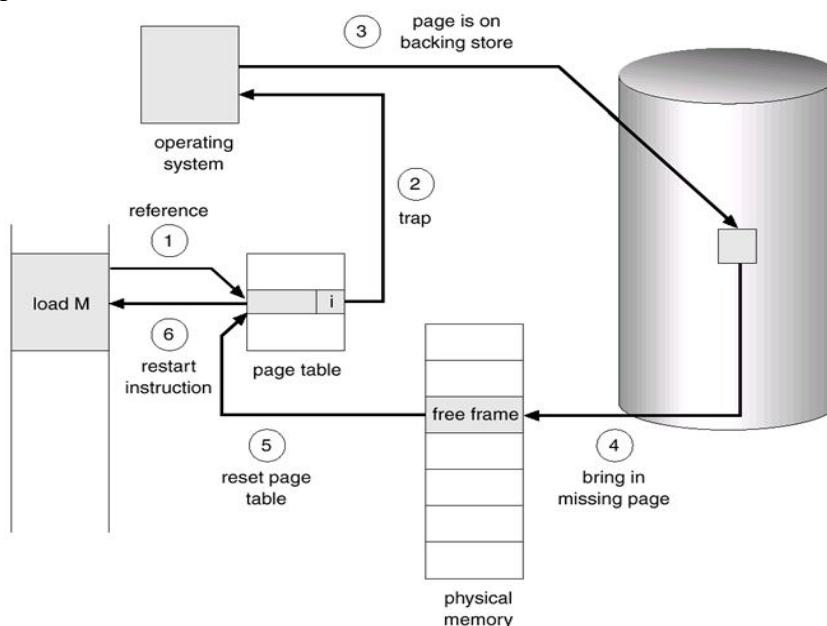
Analizirajmo sledeće dve situacije:

1. proces se izvršava ili pristupa stranici koja je u memoriji
2. proces pokušava da pristupi stranici koja nije u memoriji nego na disku



U prvom slučaju, izvršavanje programa teče normalno. U drugom slučaju, proces pristupa logičkoj stranici čija je vrednost bita validnosti i, što izaziva prekidni signal PF (*page-fault trap*). Operativni sistem tada poziva rutinu za opsluživanje, PF, čiji je zadatak da stranicu sa diska prebaci u memoriju. Prebacivanje stranice sa diska u memoriju obavlja se u više koraka :

1. referenca (load M) prouzrokovala je prekid PF, jer je prilikom čitanja stranice u tabeli detektovan *invalid bit*
2. operativni sistem poziva sistemsku rutinu za obradu, PF. Ukoliko referenca nije validna, proces se prekida jer sadrži pogrešnu instrukciju. Ukoliko je referenca validna, PF započine učitavanje stranice u memoriju
3. PF rutina pronalazi stranicu na disku, u prostoru za razmenjivanje
4. PF rutina traži slobodan okvir u fizičkoj memoriji, nakon čega prebacuje stranicu sa diska u pronađeni slobodan okvir
5. PF rutina ažurira tabelu stranica. Na ulazu koji je napravio PF prekid, upisuje se adresa okvira i poništava *i bit* (postavlja se *v bit*)
6. prekinuta instrukcija koja je uzorkovala PF prekid izvršava se iz početka , s tim što sada ima sve što joj treba u memoriji



Postupak se ponavlja za svaku stranicu procesa koja nije u memoriji, pri čemu svaki PF prekid učitava samo jednu stranicu sa diska.

DP tehnika koja učitava stranicu u memoriju isključivo posle referenciranja, pri čemu se svaki put prebacuje samo jedna stranica, naziva se čista DP tehnika (*pure demand paging*). Teorijski, jedna instrukcija bi mogla da napravi više PF prekida, ali se to retko dešava zato što se procesi uglavnom obraćaju manjem opsegu adresa, lokalizovanim u jedom delu memorijske mape.

Jedan od mogućih problema jeste nedostatak slobodnih okvira, tj slobodne fizičke memorije. PF rutina tada najčešće koristi algoritam za zamenu stranica: iz memorije se izabere jedan okvir, a stranica koja se u njemu nalazi prebaci se na disk, čime se oslobađa fizička memorija.