

SEKUNDARNE MEMORIJE

1. Posmatrajte disk sa 5000 cilindara označenih brojevima od 0-4999.
Glave za čitanje i pisanje se trenutno nalaze cilindru 143, a prethodno su bile pozicionirane na cilindru 125. Trenutni raspored reda za rad sa diskom u FIFO poretku je: 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. Odredite redosled opsluživanja disk zahteva i ukupan pomeraj (ukupan broj cilindara obuhvaćen *seek* sekvencom), ukoliko se za raspoređivanje zahteva koriste sledeći algoritmi: (a) FCFS, (b) SSTF, (c) SCAN, (d) LOOK, (e) C-SCAN, (f) C-LOOK. Za početni trenutak usvojite momenat u kome su glave diska na 143 cilindru.
 - (a) Raspored zahteva prema FCFS algoritmu je (brojevi u zagradi predstavljaju pomeraj koji glave diska prave pri prelasku na sledeći cilindar, izražen u broju cilindara):
143, 86 (57), 1470 (1384), 913 (557), 1774 (861), 948 (826), 1509 (561), 1022 (487), 1750 (728), 130 (1620).
Ukupan pomeraj glava: 7801 cilindara.
 - (b) Raspored zahteva prema SSTF algoritmu je: 143, 130 (13), 86 (44), 913 (827), 948 (35), 1022 (74), 1470 (448), 1509 (39), 1750 (241), 1774 (24).
Ukupan pomeraj glava: 1745 cilindara.
 - (c) Raspored zahteva prema SCAN algoritmu je: 143 , 913 (770), 948 (35), 1022 (74), 1470 (448), 1509 (39), 1750 (241), 1774 (24), 4999 (3225), 130 (4869), 86 (44).
Ukupan pomeraj glava: 9769 cilindara.
 - (d) Raspored zahteva prema LOOK algoritmu je: 143 , 913 (770), 948 (35), 1022 (74), 1470 (448), 1509 (39), 1750 (241), 1774 (24), 130 (486), 86 (44).
Ukupan pomeraj glava: 3319.
 - (e) Raspored zahteva prema C-SCAN algoritmu je: 143 , 913 (770), 948 (35), 1022 (74), 1470 (448), 1509 (39), 1750 (241), 1774 (24), 4999 (3225), 0 (4999), 86 (86), 130 (44).
Ukupan pomeraj glava: 9813.
 - (f) Raspored zahteva prema C-LOOK algoritmu je: 143 , 913 (770), 948 (35), 1022 (74), 1470 (448), 1509 (39), 1750 (241), 1774 (24), 86 (1688), 130 (44).
Ukupan pomeraj glava: 3363.

- 2
- a. Objasnite kako funkcionišu RAID nivoi 0 i 1.
 - b. Objasnite kako funkcionišu RAID 0+1 i RAID 1+0.
 - c. Objasnite posledice otkaza jednog diska na RAID 0+1 i RAID 1+0 sistemima.
- (a) RAID 0 (*block striping*) - stripe na nivou jednog ili više blokova podataka. Posедуje paralelizam, ali ne i redundansu. U slučaju otkaza jednog diska, svi podaci se nepovratno gube.
- RAID 1 (*disk mirroring*) - svaki disk ima svoje ogledalo. Redundansa je potpuna, ali nema paralelizma.
- (b) U kombinaciji 0+1, skup diskova deli podatke, a potom se sve stripe jedinice u celini kopiraju u svoje ogledalo. Druga kombinacija 1+0 znači da svaki disk ima svoje ogledalo a onda se podaci dele između ogledala. 1+0 teorijski je bolji u odnosu na 0+1.
- (c) Kod 0+1 u slučaju da jedan disk otkáže cela grupa stripe jedinica je neraspoloživa na 2 mesta, dok je ona u ogledalu raspoloživa. U slučaju 1+0 otkaz jednog diska utiče na pouzdanost samo te stripe jedinice koja ima samo jednu raspoloživu kopiju.

3

Data su četiri identična diska kapaciteta 10GB sa sledećim karakteristikama:

- prosečno vreme pozicioniranja (*average seek time*): 6.3msec,
- pozicioniranje s kraja na kraj (*full stroke seek*): 15msec,
- pozicioniranje na sledeći cilindar (*track to track seek*): 0.8msec,
- ugaona brzina: 7200rpm (8.33 msec),
- pristup medijumu (*sustain data rate*): 24.5MB/sec,
- propusna moć: 160MB/sec,
- srednje vreme otkaza (*mean time to failure*): 200.000 časova (23 godine),
- cena: \$50.

Od ovih diskova formiran je RAID-0 sa *stripe* jedinicom veličine 4KB (8 blokova).

a. Skicirati raspored stripe jedinicama ovog RAID sistema. Koliki je kapacitet ovog RAID-a? Koliko blokova ima ovaj RAID, a koliko stripe jedinica?

b. Odredite na kom disku se nalazi logički blok 153.

c. U redu za korišćenje diska nalaze se redom zahtevi sa sledećim veličinama transfera: 1K, 8K, 12K, 2K, 4K. Koliko zahteva može opslužiti ovaj RAID istovremeno za ovaj red, a koliko uopšte?

d. Odredite cenu po gigabajtu i stepen iskorišćenja prostora.

e. Odredite srednje vreme otkaza za ovaj RAID

(a) Sledeća tabela predstavlja raspored stripe jedinica datog sistema:

disk0	disk1	disk2	disk3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
...
N-4	N-3	N-2	N-1

Kapacitet ovog RAID-0 sistema je 40GB:

broj blokova = broj diskova · broj blokova na disku,

broj blokova = $4 \cdot (10\text{GB}/512) = 78125000$ blokova,

broj traka = broj blokova / veličina trake u blokovima,

broj traka = $78125000 / 8 = 9765625$,

- (b) Prvo se određuje kojoj traci pripada blok.

Stripe = Integer (logički blok / veličina trake) = Integer($153/8$) = 19.

Disk = (stripe % broj diskova) = moduo ($19/4$) = 3 (logički blok se nalazi na disku broj 3, a kako se diskovi numerišu počev od nule, to je četvrti disk u RAID-u).

- (c) I/O queue = (1K, 8K, 12K, 2K, 4K)

Teorijski, RAID 0 može opslužiti onoliko zahteva koliko ima diskova u RAID-u, pod uslovom da je transfer manji od veličine trake i da se trake nalaze na različitim diskovima. U ovom slučaju RAID trenutno može da izvršava najviše tri zahteva (1K, 2K, 4K), pod uslovom da se zahtevi odnose na trake smeštene na različitim diskovima. Teorijski, ovaj RAID može izvršiti 4 zahteva simultano.

- (d) Cena po GB = ukupna cena / ukupni kapacitet.

Cena po GB = $4 \times 50\$ / 4 \times 10\text{GB} = 5\$/\text{GB}$.

Stepen iskorišćenja prostora za RAID 0 je 100%.

- (e) Srednje vreme otkaza za RAID 0 se računa kao količnik srednjeg vremena otkaza za jedan disk i broja diskova u sistemu. U ovom slučaju je:

$\text{MTTF}(\text{RAID-0}) = \text{MTTF}(\text{disk}) / n$

$\text{MTTF}(\text{RAID-0}) = 200.000/4 = 50.000$ sati = oko 6 godina.

Napomena: ova vrednost ima više teorijski značaj. Za sve praktične slučajeve treba primeniti citate iz Marfijevog zakona.

4

Data su četiri identična diska kapaciteta 10GB sa sledećim karakteristikama:

- prosečno vreme pozicioniranja (*average seek time*): 6.3msec,
- pozicioniranje s kraja na kraj (*full stroke seek*): 15msec,
- pozicioniranje na sledeći cilindar (*track to track seek*): 0.8msec,
- ugaona brzina: 7200rpm (8.33 msec),
- pristup medijumu (*sustain data rate*): 24.5MB/sec,
- propusna moć: 160MB/sec,
- srednje vreme između otkaza: 200.000 sati (23 godine),
- cena: \$50.

Od ovih diskova formiran je RAID-1.

a. Skicirati ovaj RAID sistem. Koliki je kapacitet i koliko logičkih blokova ima ovaj RAID?

b. Odredite na kom disku se nalazi logički blok 335.

c. U redu za korišćenje diska nalaze se redom zahtevi sa sledećim veličinama transfera: 1K, 8K, 12K, 2K, 4K. Koliko zahteva može opslužiti ovaj RAID istovremeno za ovaj red, a koliko uopšte?

d. Odredite cenu po gigabajtu.

(a) Sledeća tabela ilustruje dati RAID:

disk0	disk1	disk2	disk3
0	0	$N/2$	$N/2$
1	1	$N/2+1$	$N/2+1$
2	2	$N/2+2$	$N/2+2$
3	3	$N/2+3$	$N/2+3$
4	4	$N/2+4$	$N/2+4$
5	5	$N/2+5$	$N/2+5$
...
$N/2-1$	$N/2-1$	N-1	N-1

Kapacitet ovog RAID-1 sistema je 20GB:

broj blokova = (broj diskova / 2) · broj blokova na disku,

broj blokova = $(4/2) \cdot (10GB/512) = 39062500$ blokova,

- (b) Logički blok 335 nalazi se u disku 0 (na kome se nalaze logički blokovi 0-19.531.249), a njegova kopija na disku 1.
- (c) I/O queue = (1K, 8K, 12K, 2K, 4K)

Teorijski, RAID 1 može opslužiti onoliko zahteva koliko parova diskova ima u RAID-u, pod uslovom da se zahtevi odnose na različite diskove. U ovom slučaju RAID trenutno može da izvršava najviše dva zahteva, pod uslovom da se zahtevi odnose na različite diskove.

- (d) Cena po GB = ukupna cena / ukupni kapacitet.

$$\text{Cena po GB} = 4 \times 50\$ / 2 \times 10\text{GB} = 10\$/\text{GB}.$$

Stepen iskorišćenja prostora za RAID 1 je 50%.

5 Data su četiri identična diska kapaciteta 10GB, sa sledećim karakteristikama:

- prosečno vreme pozicioniranja (*average seek time*): 6.3msec,
- pozicioniranje s kraja na kraj (*full stroke seek*): 15msec,
- pozicioniranje na sledeći cilindar (*track to track seek*): 0.8msec,
- ugaona brzina: 7200rpm (8.33 msec),
- pristup medijumu (*sustain data rate*): 24.5MB/sec,
- propusna moć: 160MB/sec,
- cena: \$50.

Od ovih diskova formiran je RAID-3 sa *stripe* jedinicom veličine 1B.

a. Skicirati ovaj RAID sistem. Koliki je kapacitet, koliko logičkih blokova a koliko *stripe* jedinica ima ovaj RAID?

b. Odredite na kom disku se nalazi logički blok 21.411.

c. U redu za korišćenje diska nalaze se redom zahtevi sa sledećim veličinama transfera: 1K, 8K, 12K, 2K, 4K. Koliko zahteva može opslužiti ovaj RAID istovremeno za ovaj red, a koliko uopšte?

d. Odredite cenu po gigabajtu i stepen iskorišćenja prostora..

- (a) Sledeća tabela ilustruje dati RAID (trake veličine 1B):

disk0	disk1	disk2	disk3
0	1	2	P
4	5	6	P
6	7	8	P
9	10	11	P
12	13	14	P
15	16	17	P
...
N-3	N-2	N-1	P

Kapacitet ovog RAID-1 sistema je 30GB.

broj blokova = (broj diskova - 1) · broj blokova na disku,

broj blokova = (4-1) · (10GB/512) = 58593750 blokova,

broj traka = broj blokova · broj traka u bloku,

broj traka = 58593750 · 512 = 30.000.000.000.

- (b) Logički blok LA=21411 nalazi se na sva tri diska.

Početak je na prvom disku: $(LA \cdot 512) \% (N-1) = (21411 \cdot 512) \% 0 = 0$.

Početak je u $LA/(N-1) = 21411/3 = 7137$ bloku prvog diska.

- (c) Kako je veličina trake 1B, RAID 3 može da izvršava samo jedan zahtev u jednom trenutku vremena.

- (d) Cena po GB = ukupna cena / ukupni kapacitet.

Cena po GB = $4 \times 50\$ / 3 \times 10GB = 6.66\$/GB$.

Stepen iskorišćenja prostora za RAID 3 je 75%.

6 Data su četiri identična diska kapaciteta 10GB sa sledećim karakteristikama:

- prosečno vreme pozicioniranja (*average seek time*): 6.3msec,
- pozicioniranje s kraja na kraj (*full stroke seek*): 15msec,
- pozicioniranje na sledeći cilindar (*track to track seek*): 0.8msec,
- ugaona brzina: 7200rpm (8.33 msec),
- pristup medijumu (*sustain data rate*): 24.5MB/sec,
- propusna moć: 160MB/sec,
- cena: \$50.

Od ovih diskova formiran je RAID-4 sa *stripe* jedinicom veličine 2KB.

- a. Skicirati ovaj RAID sistem. Koliki je kapacitet, koliko logičkih blokova, a koliko *stripe* jedinica ima ovaj RAID?
- b. Odredite na kom se disku se nalazi logički blok 3.945.
- c. U redu za korišćenje diska nalaze se redom zahtevi sa sledećim veličinama transfera: 1K, 8K, 12K, 2K, 4K. Koliko zahteva može opslužiti ovaj RAID istovremeno za ovaj red, a koliko uopšte?
- d. Odredite cenu po gigabajtu i stepen iskorišćenja prostora..

- (a) Sledeća tabela ilustruje dati RAID (trake veličine 2KB):

disk0	disk1	disk2	disk3
0	1	2	P
3	4	5	P
6	7	8	P
9	10	11	P
12	13	14	P
15	16	17	P
...
N-3	N-2	N-1	P

Kapacitet ovog RAID-1 sistema je 30GB:

broj blokova = (broj diskova - 1) · broj blokova na disku,

broj blokova = (4-1) · (10GB/512) = 58593750 blokova,

broj traka = broj blokova / veličina trake u blokovima,

broj traka = 58593750 / 4 = 146.484.375.

- (b) Prvo se određuje kojoj traci pripada blok.

Stripe = Integer (logički blok / veličina trake) = Integer(3945/4) = 986.

Disk = (stripe % (broj diskova-1)) = moduo (986/3) = 2 (logički blok se nalazi na disku broj 2, a kako se diskovi numerišu počev od nule to je treći disk u RAID-u).

- (c) I/O queue = (1K, 8K, 12K, 2K, 4K).

Teorijski, RAID 4 može opslužiti onoliko zahteva koliko ima diskova u RAID-u, umanjeno za 1, pod uslovom da je transfer manji od veličine trake i da se trake nalaze na različitim diskovima. U ovom slučaju RAID trenutno može da izvršava najviše dva zahteva (1K, 2K), pod uslovom da se zahtevi odnose na trake smeštene na različitim diskovima. Teorijski, ovaj RAID može izvršiti 3 zahteva simultano.

- (d) Cena po GB = ukupna cena / ukupni kapacitet.

Cena po GB = $4 \times 50\$ / 3 \times 10\text{GB} = 6.66\$/\text{GB}$.

Stepen iskorišćenja prostora za RAID 4 je 75%.

7

Data su četiri identična diska kapaciteta 10GB, sa sledećim karakteristikama:

- prosečno vreme pozicioniranja (*average seek time*): 6.3msec,
- pozicioniranje s kraja na kraj (*full stroke seek*): 15msec,
- pozicioniranje na sledeći cilindar (*track to track seek*): 0.8msec,
- ugaona brzina: 7200rpm (8.33 msec),
- pristup medijumu (*sustain data rate*): 24.5MB/sec,
- propusna moć: 160MB/sec,
- srednje vreme između otkaza: 200.000 sati (23 godine),
- cena: \$50.

Od navedenih diskova formiran je RAID-5 sa *stripe* jedinicom veličine 4KB (8 blokova).

a. Skicirati ovaj RAID sistem. Koliki je kapacitet, koliko logičkih blokova, a koliko *stripe* jedinica ima ovaj RAID?

b. Odredite na kom disku se nalazi logički blok 111.

c. U redu za korišćenje diska nalaze se redom zahtevi koji imaju sledeće veličine transfera: 1K, 8K, 12K, 2K, 4K. Koliko zahteva može opslužiti ovaj RAID istovremeno za ovaj red, a koliko uopšte?

d. Odredite cenu po gigabajtu i stepen iskorišćenja prostora..

- (a) Sledeća tabela ilustruje dati RAID (trake veličine 2KB):

disk0	disk1	disk2	disk3
0	1	2	P
3	4	P	5
6	P	7	8
P	9	10	11
12	13	14	P
15	16	P	17
...
N-3	N-2	N-1	P

Kapacitet ovog RAID-1 sistema je 30GB:

broj blokova = (broj diskova - 1) · broj blokova na disku,

broj blokova = (4-1) · (10GB/512) = 58.593.750 blokova

broj traka = broj blokova / veličina trake u blokovima,

broj traka = 58.593.750 / 8 = 7.324.218.

- (b) Prvo se određuje kojoj traci pripada blok.

Stripe = Integer (logički blok / veličina trake) = Integer(111/8) = 13.

Iz prethodne tabele se vidi da se traka br.13 nalazi na disku 1.

- (c) I/O queue = (1K, 8K, 12K, 2K, 4K).

Teorijski, RAID 5 može opslužiti onoliko zahteva koliko ima diskova u RAID-u, umanjeno za 1, pod uslovom da je transfer manji od veličine trake i da se trake nalaze na različitim diskovima. U ovom slučaju RAID trenutno može da izvršavaju najviše tri zahteva (1K, 2K, 4K), pod uslovom da se zahtevi odnose na trake smeštene na različitim diskovima. Teorijski, ovaj RAID može izvršiti 3 zahteva simultano.

- (d) Cena po GB = ukupna cena / ukupni kapacitet.

Cena po GB = 4 x 50\$ / 3 x 10GB = 6.66\$/GB.

Stepen iskorišćenja prostora za RAID 5 je 75%.

SISTEMI DATOTEKA

1 Koje se operacije mogu izvršavati nad datotekama?

- kreiranje datoteke,
- čitanje podataka iz datoteke u memorijski bafer,
- upis podataka u datoteku,
- pozicioniranje unutar datoteke (*file seek*),
- brisanje datoteke,
- odsecanje datoteke (*truncate*).

Rad sa datotekama se može ubrzati uvođenjem operacija otvaranja (*open*) i zatvaranja (*close*) datoteka, koje uz pomoć odgovarajućih memorijskih struktura ubrzavaju pristup datotekama.

2 Šta je (a) logička a šta (b) fizička struktura datoteke?

- (a) Logička struktura datoteke je način na koji je datoteka predstavljena korisniku. U najjednostavnijem slučaju, datoteka nema svoju logičku strukturu i predstavljena je kao kolekcija reči, odnosno bajtova. U jednostavnije logičke strukture spadaju strukture zapisa, pri čemu jedan zapis u datoteci može biti fiksne ili promenljive dužine. Primer zapisa je linija u tekstualnoj datoteci. Složenije strukture predstavljaju formatirani dokumenti.
- (b) Pod fizičkom strukturom datoteke podrazumeva se način smeštanja datoteke na masovnim memorijskim medijumima, odnosno prostorni raspored datoteke u sistemu datoteka.

3 Šta je tip datoteke, a šta ekstenzija?

Pomoću tipa, operativni sistem može preliminarno da odredi vrstu datoteke i da je poveže sa nekom aplikacijom. Tip datoteke se može realizovati produženjem (ekstenzijom) imena datoteke, što je slučaj za DOS/Windows operativne sisteme.

4 Nabrojite značajnije tipove datoteka, zavisno od namene, na DOS i Windows operativnim sistemima i podrazumevane ekstenzije dodeljene tim tipovima datoteka.

- binarna izvršna datoteka (.exe, .com),
- prevedeni izvorni kod (.obj),
- statičke i dinamičke biblioteke (.lib, .dll),
- izvorni kod (c, cpp, asm, pas),
- batch izvršna datoteka (.bat),
- tekstualna datoteka (.txt),
- tekstualni dokument kreiran tekst procesorom (.rtf, .doc, .sxw, .odt),
- arhive (.zip, .rar, .arj),
- rasterske slike (.jpeg, .bmp, .gif),
- multimedijalni sadržaji (.mp3, .wav, .wma, .avi, .mpeg).

5 Objasnite sledeće metode pristupa datotekama: (a) direktan (b) sekvencijalni (c) pristup pomoću indeksne datoteke. Navedite primere kada aplikacija datoteci pristupa (d) sekvencijalno, (e) direktno.

- (a) Informacije se prosleđuju u tačnom redosledu, jedna iza druge, uvek u odnosu na vrednost tekućeg ukazivača. Posle svakog pristupa datoteci, vrednost tekućeg ukazivača se ažurira. Sekvencijalni pristup zahteva da postoji mogućnost da se datoteka premota na početak, tako da vrednost tekućeg pokazivača bude nula.
- (b) Omogućava pristup bilo kom delu datoteke, tako što se najpre odredi njegova pozicija na disku, a zatim pristupi podacima. Direktan pristup omogućava korisniku da pristupi krajnjem bloku datoteke bez čitanja prethodnog sadržaja.
- (c) Svakoј datoteci (tabeli) pridružena je indeksna datoteka, uređena po nekom kriterijumu, pomoću koje se prilikom čitanja brzo može naći odgovarajući zapis. Prilikom upisa novog sloga u datoteku, ažurira se i indeksna datoteka.
- (d) Štampanje sadržaja datoteke.
- (e) Izmena sadržaja n-tog zapisa.

6 a. Šta je direktorijum?

b. Koje se operacije mogu izvršiti nad direktorijumom?

- (a) Direktorijum je struktura koja sadrži kontrolne blokove svih datoteka logički smeštene u njemu.
- (b) Operacije koje se mogu izvesti nad direktorijumima su: prikazivanje sadržaja direktorijuma ("listanje" direktorijuma), pretraživanje direktorijuma, promena imena datoteke, kreiranje i brisanje datoteka i poddirektorijuma u okviru tekućeg direktorijuma.

7 Opišite: (a) jednonivovsku strukturu direktorijuma, (b) dvonivovsku strukturu direktorijuma, (c) direktorijumsko stablo.

d. Kako se na jednonivovskom sistemu koji podržava duga imena datoteka može simulirati direktorijumsko stablo?

- (a) Na sistemu datoteka postoji jedan direktorijum u kome se nalaze sve datoteke.
- (b) Na prvom nivou se nalazi glavni direktorijum (MFD - master file directory), a na drugom poseban direktorijum za svakog korisnika (UFD - user file directory).
- (c) Direktorijum najvišeg nivoa je početni ili korenski direktorijum (root). U svakom direktorijumu se mogu kreirati datoteke i poddirektorijumi (subdirectory) koji predstavljaju grane tog stabla.
- (d) Korišćenjem specijalnih karaktera, kao što je tačka. Na primer, notacija temp.images.pic1 bi ukazivala na datoteku pic1 u poddirektorijumu images direktorijuma temp. Napomena: ovaj način simulacije nije ostvarljiv na sistemima datoteka na kojima je dužina imena datoteke ograničena na npr. 7 ili 8 karaktera.

8 Šta je apsolutna a šta relativna putanja datoteke? Navedite primer apsolutne i relativne putanje na UNIX sistemima.

Putanja datoteke je apsolutna, ako je izražena u odnosu na početni direktorijum (npr. /home/jonhsmith), a relativna ako je izražena u odnosu na bilo koji drugi direktorijum, osim početnog (npr. backup/disk1).

- 9 a. Šta je hard a šta simbolički link na UNIX sistemima ?
- b. Koji od prethodno pomenutih linkova mogu ukazivati na nepostojeće objekte sistema datoteka i direktorijume ?
- (a) Hard link je alternativno ime datoteke, odnosno alternativni ukazivač na i-node datoteke. Simbolički link je prečica ka objektu u sistemu datoteka, odnosno zaseban objekat sistema datoteka koji koristi jedan i-node i jedan blok podataka u kom je zapisana lokacija originalnog objekta.
- (b) Simbolički linkovi i prečice.

10 Šta je sistem datoteka i od kojih delova je sastavljen? Koji delovi predstavljaju premašenje sistema?

Sistem datoteka je skup metoda i struktura podataka koje operativni sistem koristi za čuvanje datoteka. Sistem datoteka čine: zaglavlje, meta podaci (strukture za organizaciju podataka na medijumu) i sami podaci, odnosno datoteke i direktorijumi. Zaglavlje i meta-podaci čine premašenje sistema, ali bez njih sistem datoteka ne može da funkcioniše.

- 11 a. Koje su strukture podataka definisane u operativnoj memoriji radi ubrzavanja rada sa sistemom datoteka?
- b. Kako se pristupa datoteci putem ovih struktura?
- (a) Tabela otvorenih datoteka na sistemskom nivou i tabela otvorenih datoteka po procesu.
- (b) Prilikom otvaranja datoteke kontrolni blok se iz direktorijuma upisuje u glavnu memorijsku tabelu. Prilikom svakog pristupa datoteci, iz tabele procesa se preko glavne tabele otvorenih blokova pronalaze blokovi datoteke na disku, a zatim se njima pristupa.

12 Posmatrajte datoteku veličine 100 blokova, čiji se kontrolni blok nalazi u memoriji (u slučaju indeksne alokacije, pretpostaviti da je i indeksni čvor u memoriji računara). Odredite koliko je disk I/O operacija potrebno za izvršenje sledećih operacija na sistemima organizovanim pomoću metoda kontinualne alokacije, vezivanja blokova i indeksnih čvorova. Pretpostavite da se u slučaju kontinualne alokacije datoteka može nastaviti samo sa kraja i da se blok, koji sadrži informacije za dodavanje u datoteku, nalazi u memoriji.

- a. Blok se dodaje na početak datoteke.
- b. Blok se dodaje na sredinu datoteke.
- c. Blok se dodaje na kraj datoteke.
- d. Blok se uklanja sa početka datoteke.
- e. Blok se uklanja iz sredine datoteke.
- f. Blok se uklanja sa kraja datoteke.

	Kontinualna alokacija	Vezivanje blokova	Metoda indeksnih čvorova
(a)	201	1	1
(b)	101	52	1
(c)	1	3	1
(d)	198	1	0
(e)	98	52	0
(f)	0	100	0

- 13** a. Koji je osnovni nedostatak metode dodele kontinualnog prostora?
- b. Kakav je pristup datoteci ukoliko se za organizaciju sekundarne memorije koristi dodela kontinualnog prostora?

- (a) Najveći problem ove metode je to što se datoteci odmah mora dodeliti ukupan adresni prostor na disku. Posle toga, datoteka ne može da raste ukoliko posle nje nema dovoljno kontinualnog slobodnog prostora.
- (b) Pristup datoteci je direktan.

- 14** a. Koji način organizacije koristi FAT sistem datoteka?

- (a) FAT sistem datoteka predstavlja kombinovanu metodu kontinualne dodele prostora na nivou extent-a i mape datoteka. Sistem datoteka se prilikom kreiranja deli na extent-e, koji se nazivaju klasteri (*cluster*). U okvirima jednog klastera datoteci je dodeljen kontinualan prostor. Datoteci se može dodeliti ceo broj klastera u proizvoljnom rasporedu, ali jedan klaster ne mogu koristiti dve datoteke. Svi klasteri jedne datoteke čine njenu povezanu listu. Povezane liste svih datoteka čuvaju se u FAT tabeli (koja predstavlja mapu datoteke).

- 15** a. Na koji se način može razrešiti problem adresiranja velikih datoteka na sistemima koji su organizovani pomoću metode indeksnih čvorova?
- b. Kako ovaj problem razrešava UNIX?
- (a) Vezivanjem indeksnih blokova (za opis prostornog rasporeda datoteke koristi se veći broj indeksnih blokova) i višenivovskim adresiranjem (kontrolni blok datoteke ukazuje na jedan indeksni blok prvog nivoa, koji ukazuje na n indeksnih blokova drugog nivoa; indeksni blokovi drugog nivoa adresiraju blokove datoteke; broj nivoa se po potrebi može povećati).
- (b) UNIX razrešava problem tronivovskim adresiranjem blokova podataka.
- 16** Posmatrajte sistem koji podržava metode dodele kontinualnog prostora, vezivanja blokova i indeksnih čvorova. Na osnovu kog kriterijuma treba odlučiti koja je strategija optimalna za neki sistem datoteka?

Na osnovu prosečne veličine datoteka i metode pristupa datotekama. U slučaju da su datoteke relativno male i da im se najčešće pristupa sekvencijalno, koristi se dodela kontinualnog prostora. U slučaju da su datoteke velike i da im se, najčešće, pristupa sekvencijalno, koristi se metoda vezivanja blokova. U slučaju da su datoteke velike i da im se najčešće pristupa direktno, koristi se metoda indeksnih blokova.

- 17** a. Koje se metode koriste za upravljanje slobodnim prostorom?
- b. Zašto se mape bitova moraju čuvati na disku, a ne u operativnoj memoriji?
- (a) i. Mape bitova - za svaki blok diska uvodi se bit koji vrednostima 0 i 1 opisuje da li je taj blok slobodan ili zauzet. Mapa bitova se formira kao struktura podataka na disku, odnosno kao niz svih prethodno navedenih bitova.
- ii. Povezane liste - početak liste ukazuje na prvi slobodni blok, a svaki slobodni blok ukazuje na sledeći.
- (b) Ukoliko se čuva na disku, mapa bitova neće biti izgubljena u slučaju kraha sistema (na primer, otkaz memorije ili nestanak struje).

18 a. Koji su osnovni delovi jednog UNIX sistema datoteka?

b. Šta sve sadrži jedan indeksni čvor (*i-node*)?

(a) zaglavlje (*superblock*), tabela indeksnih čvorova (*i-node tabela*), blokovi sa podacima (*data blocks*), direktorijumski blokovi (*directory blocks*) i blokovi indirektnih pokazivača (*indirection block*),

(b) tip objekta i pristupna prava za tri vlasničke kategorije, broj hard linkova na dati objekat, UID vlasnika, GID, veličina objekta izraženu u bajtovima, vreme zadnjeg pristupa objektu, vreme zadnje modifikacije objekta, vreme zadnje modifikacije indeksnog čvora objekta i lista direktnih i indirektnih pokazivača na blokove sa podacima.

19 Na FAT32 sistemu datoteka, sa klasterima veličine 2KB, nalazi se datoteka veličine 9.123 bajtova. Datoteka je redom smeštena u sledećim klasterima: 5, 10, 30, 40, 75.

a. Koliko blokova veličine 512 bajtova datoteka zauzima, a koliko blokova koristi?

b. Kolika je interna fragmenacija za ovu datoteku?

(a) Datoteka zauzima 5 klastera, a to je $5 \cdot 2\text{KB} / 512\text{B} = 20$ blokova .

Datoteka koristi $9123 / 512 = 17,8 = 18$ blokova, pri čemu je neiskorišćeno 94 bajta 18-tog bloka.

(b) Interna fragmentacija: $5 \cdot 2048 - 9123 = 1117$ bajtova.

20 Dat je UNIX sistem datoteka sa veličinom sistemskog bloka 1KB i indeksnim čvorovima u kojima se nalazi 10 direktnih pokazivača, jedan indirektni, jedan dupli indirektni i jedan trostruki indirektni pokazivač. U sistemu datoteka nalazi se datoteka veličine 7.899 bajtova, čiji je niz direktnih pokazivača u indeksnom čvoru: 2, 12, 55, 60, 76, 80, 111, 321, free, free.

a. Koliko blokova diska veličine 512B datoteka zauzima, a koliko efektivno koristi?

b. Kolika je interna fragmenacija za ovu datoteku?

c. Odredite najveću veličinu datoteke koja se može adresirati samo direktnim pokazivačima.

d. U kom bloku područja podataka se nalazi 3.144-ti bajt datoteke? Pretpostavite da su direktni pokazivači, sistemski blokovi, blokovi u području podataka i bajtovi datoteke numerisani od 0 do N-1.

- (a) Datoteka zauzima 8 sistemskih blokova, odnosno $8 \cdot 1\text{KB} = 8\text{KB}$, što je ukupno 16 blokova diska.

Datoteka koristi $7.899/512 = 15,4 = 16$ blokova, pri čemu je neiskorišćeno 293 bajta 16-tog bloka.

- (b) Interna fragmentacija: $8 \cdot 1024 - 7899 = 239$ bajtova.

- (c) Najveća veličina datoteke = broj direktnih pokazivača · veličina sist. bloka

Najveća veličina datoteke = $10 \cdot 1\text{KB} = 10\text{KB}$

- (d) $3144/512 = 6,14 = 7$ - Traženi bajt se nalazi u sedmom bloku datoteke (blok broj 6 kad se broji od nule)

Najpre se određuje koji direktni pokazivač upućuje na šesti blok:

$DP = \text{blok datoteke} / \text{veličina sistemskog bloka}$,

sistemski blok = 2 bloka diska,

$DP = 6/2 = 3$ - bloku odgovara pokazivač br 3. (četvrti po redu u indeksnom čvoru). Četvrti pokazivač ukazuje na sistemski blok $SB = 60$,

$\text{offset} = 6\%2 = 0$ (blok 0, odnosno prvi blok, u tom sistemskom bloku 60),

Blok (područje sa podacima) = $SB \cdot \text{veličina SB} + \text{offset}$,

veličina SB = $1\text{KB} = 2$ bloka na disku,

Blok (područje sa podacima) = $60 \cdot 2 + 0 = 120$.

- 21** Dat je UNIX sistem datoteka sa veličinom sistemskog bloka 8KB i indeksnim čvorovima u kojima se nalazi 10 direktnih pokazivača, jedan indirektni, jedan dupli indirektni i jedan trostruki indirektni pokazivač. U sistemu datoteka nalazi se datoteka veličine 22.045 bajtova, čiji je niz direktnih pokazivača u indeksnom čvoru: 7, 22, free, free, free, free, free, free.

a. Koliko blokova diska veličine 512B datoteka zauzima, a koliko efektivno koristi?

b. Kolika je interna fragmenacija za ovu datoteku?

c. Odredite najveću veličinu datoteke koja se može adresirati samo direktnim pokazivačima.

d. U kom bloku područja podataka se nalazi 15.600-ti bajt datoteke? Pretpostavite da su direktni pokazivači, sistemski blokovi, blokovi u području podataka i bajtovi datoteke numerisani od 0 do N-1.

- (a) Datoteka zauzima 3 sistemska bloka, odnosno $3 \cdot 8\text{KB} = 24\text{KB}$, što je ukupno 48 blokova diska.

Datoteka koristi $22.045/512 = 43,05 = 44$ blokova, pri čemu je neiskorišćeno 483 bajta 44-tog bloka.

- (b) Interna fragmentacija: $3 \cdot 8192 - 22.045 = 2.531$ bajtova.

- (c) Najveća veličina datoteke = broj direktnih pokazivača · veličina sist. bloka.

Najveća veličina datoteke = $10 \cdot 8\text{KB} = 80\text{KB}$.

- (d) $15600/512 = 30.1 = 31$ - Traženi bajt se nalazi u 31. bloku datoteke (blok broj 30 kad se broji od nule).

Najpre se određuje koji direktni pokazivač upućuje na blok 30:

$DP = \text{blok datoteke} / \text{veličina sistemskog bloka},$

sitemski blok = 2 bloka diska,

$DP = 30/16 = 1$ - bloku odgovara pokazivač br 1. (drugi po redu u indeksnom čvoru). Drugi pokazivač ukazuje na sistemski blok $SB = 22$.

$\text{offset} = 30 \% 16 = 14$ (blok 14 u sistemskom bloku 22),

Blok (područje sa podacima) = $SB \cdot \text{veličina SB} + \text{offset},$

veličina SB = $8\text{KB} = 16$ bloka na disku,

Blok (područje sa podacima) = $22 \cdot 16 + 14 = 366$.