

Kolik minimálně uzluu bude mít tento strom, jestliže popisuje soubor o 6 blocích(neuvažujeme extenty)? Zdůvodnit. Uvést další účely použití B+ stromuu(případne jejich různé varianty) v souborových systémech.

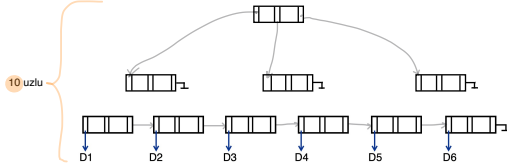
- 4 uzly

a) viz zadani

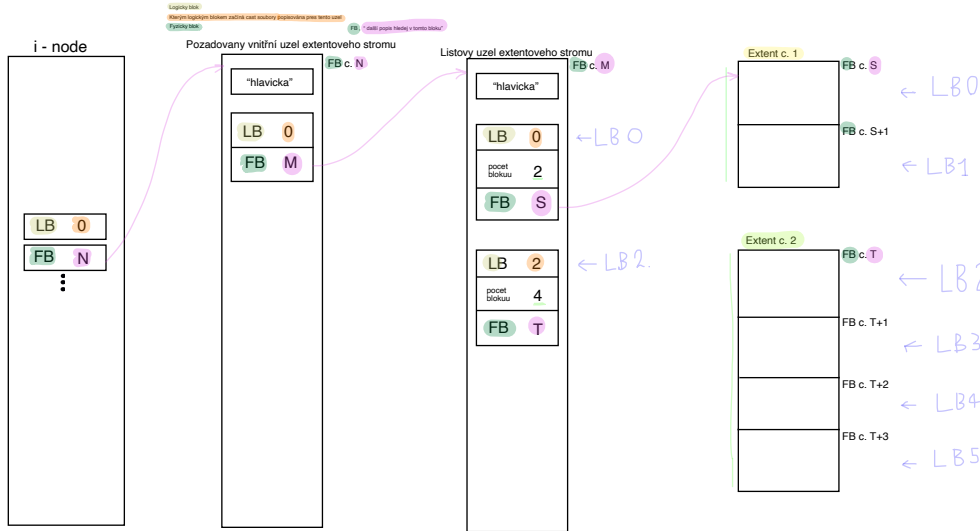
- Na listové úrovni lze z jednoho listu odkazovat max. 2 datové bloky, neboť 3. odkaz je použit pro zřetězení seznamu.
- Jsou tedy zapotřebí 3 listy.
- Z vnitřního uzlu lze odkazovat až 3 podřízené uzly. Kostce tedy 1 nadřazený uzel k třem listům
- Ten bude již kořenem.

- Pouze jeden.
- Je-li disk malo fragmentovany, 6 datovych blokuu lze uložit do jednoho extentu a z B+ strumu tak stačí použít jeden uzel, který bude koren.

- 10
- minimalni zaplitenost listu je $(\text{int}) (3/2)$
- minimalni zaplitenost vnitrih uzlu = $(\text{int}) (3/2 + 0.5)$



- Zrychluje své práce a velikosti soudby;
- mění šepo vyvířené indoevropských struktury;
- mění objem metafát, které je třeba procházet a učitovat;
- lepší lokalitu metafát a dat



```
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/stat.h>

int main()
{
    int fd, sum, i;
    char a[] = { 0, 1 };
    char b[] = { 2, 3 };
    char c[] = { 28, 28, 28, 28, 28, 28 };

    // 0_RDONLY - otevrim pro zapsi
    // O_CREAT - pokud neexistuje, bude vytvoren
    // O_TRUNC - pokud existuje, bude zkraslen na 0
    // S_IRUSR, S_IWUSR - pristupova prava
    fd = open( "/tmp/x", O_RDONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR );

    // z pole a zapisu do souboru 2 byty
    write( c, fd, 2 );

    // posumu se o 1 byte oproti aktualni pozici, vznikla ridky soubor.
    lseek( c, fd, 1, SEEK_CUR );

    // zapisu dalsi 2 byty z b
    write( c, fd, 2 );
    close( c, fd );

    // fd = 3, viz tabulka deskriptoru (0 - stdin, 1 - stdout, 3 - fp)
    // ----- Pouzetecku hodnotke je 3, protoze se jedna o 1. vyhny popisovac po standardnich popisovacich stdin, stdout, stderr
    sum = fd = open( "/tmp/x", O_RDONLY );

    // ----- Read sice chce nacist 6 bytuu, ale v souboru je jen 5 bytuu, a proto vrati 5, dale se prepise prvich 5 polozek v poli c[]
    sum += read( fd, c, 6 );
    close( fd );

    // ----- Prvni 2 byty soubosu jsou z pole a[], 3. nebyl nacten(==0) - ridky soubor, 4. a 5. byte jsou z pole b[], posledni nebyl preprany
    for( i = 0; i < 6; i++) sum += c[i];

    // 42
    printf( "the sum is %d\n", sum );
    return 0;
}
```