

Tema 3 - Structuri de date - seria CB - 2019

Linux Filesystem

Responsabili temă	Daniel Nicolescu, Ioana Alexandru, Alexandru Plăcintă
Data publicării	25.04.2019
Termen predare	19.05.2019 (ora 23:55) Se acceptă teme trimise cu penalizare de 10 puncte/zi (din maxim 100 puncte) până la data de 21.05.2019 (ora 23:55)
Versiune document	1

Abstract

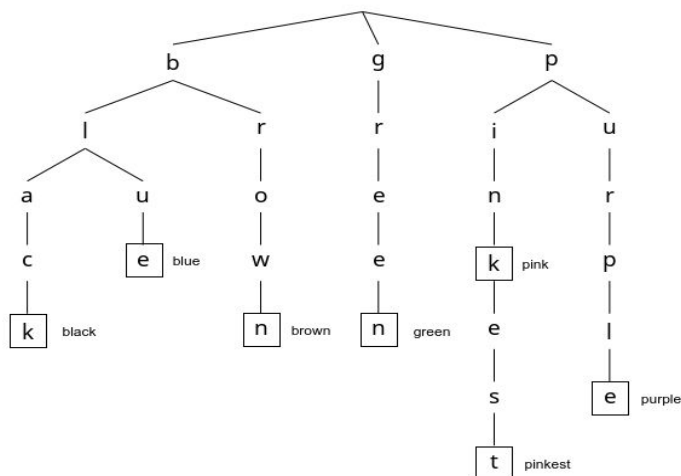
Un arbore multi căi este un arbore în care fiecare nod poate avea oricâți descendenți.

Un arbore de prefixe (trie) este un tip de arbore de căutare cu următoarele proprietăți:

- Fiecare nod poate avea între 0 și $|\Sigma|$ descendenți ($|\Sigma|$ reprezintă dimensiunea alfabetului)
- Spre deosebire de un arbore binar de căutare, unde fiecărui nod i se asociază o cheie, la arborele de prefixe cheile se asociază căii până la nod

Exemplu:

Fie cuvintele brown, blue, black, pink, purple, pinkest și green. Acestea generează următorul trie:



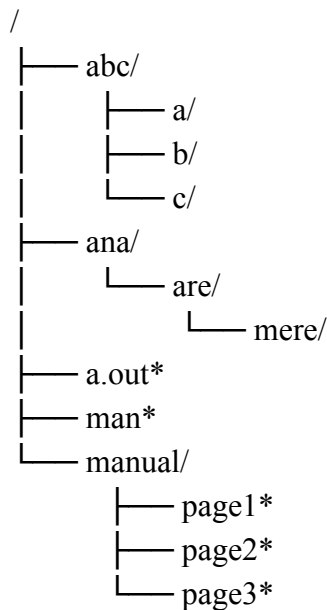
Cerință

Tema constă în implementarea unui sistem de fișiere virtual și a unui set de comenzi de terminal asemănătoare celor din Linux ce lucrează cu acesta. Sistemul de fișiere va fi reprezentat sub forma unui arbore de tip trie, în care fiecare nod va fi un caracter din numele căii. Puteți nota căile de fișier valide oricum doriți, cât timp păstrați structura de trie.

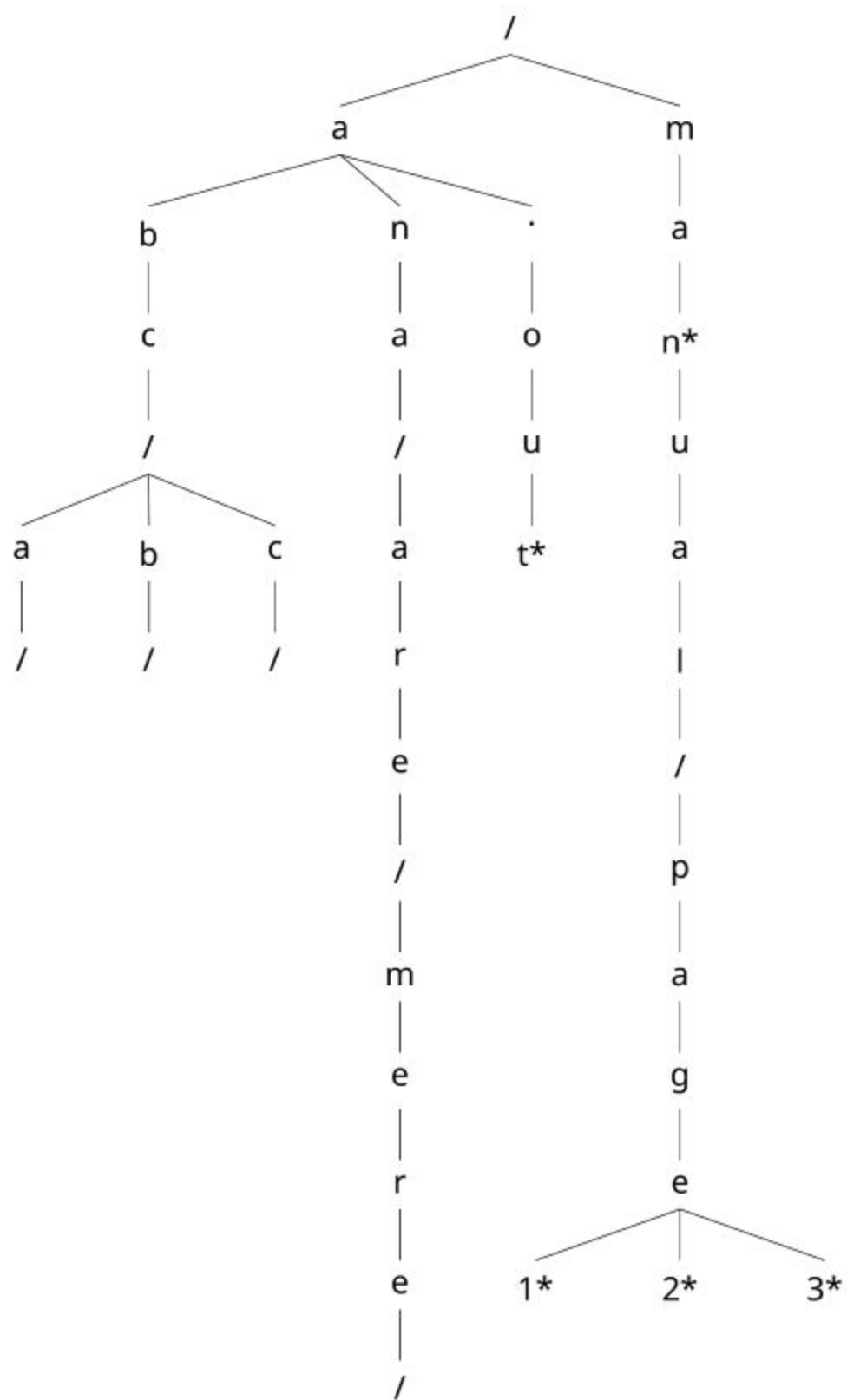
Se recomandă ca fiii unui nod să fie păstrați în ordine alfabetică într-o listă de pointeri către nod (atenție: structura pentru nod va trebui pre-declarată în acest caz). O abordare în care toate nodurile au 26 de fii alocați (deși în realitate sunt folosiți doar 2-3) va duce rapid la umplerea memoriei (un astfel de arbore având 26^n noduri alocate, unde n este adâncimea arborelui).

Exemplu

Fie următoarea structură de fișiere:



Aceasta va genera următorul arbore de prefixe (frunzele din trie marcate cu * marchează căile de fișier valide - similar cu încadrarea în pătrat din schema de mai sus).



Comenzi

Programul citește comenzile de la tastatură, acționând asupra lor atunci când se apasă tasta `Enter` (este citit caracterul `newline`). Se vor implementa următoarele comenzi:

Comenzi navigare sistem de fișiere

- **pwd**
 - afișează calea absolută a directorului curent
- **ls [-F]** [nume_director]
 - afișează fiii (fișierele și directoarele) directorului dat ca parametru sau a directorului curent în cazul în care nu se specifică `nume_director`; afișarea se va face în ordine lexicografică, pe o singură linie, cu câte un spațiu între fii
 - dacă este folosită opțiunea `-F`, concatenează caracterul `'/'` la sfârșitul numelor directoarelor, respectiv caracterul `'*'` la sfârșitul numelor fișierelor
- **cd** nume_director
 - schimbă directorul curent cu cel dat ca parametru

Comenzi creare fișiere și directoare

- **mkdir** nume_director1 [nume_director2, ...]
 - creează unul sau mai multe directoare cu numele date ca parametru
- **touch** nume_fisier1 [nume_fisier2, ...]
 - creează unul sau mai multe fișiere cu numele date ca parametru
- **rmdir** nume_director1 [nume_director2, ...]
 - șterge directoarele goale cu numele date ca parametru
- **rm** cale1 [cale2, ...]
 - șterge fișierele sau directoarele date ca parametru și conținutul acestora (comportamentul default este similar comenzii `rm -r` din Linux)
- **mv** nume_fisier1 [nume_fisier2, ...] noua_cale
 - mută fișierele date ca parametru la calea `noua_cale` (ultimul parametru)
- **cp** nume_fisier1 [nume_fisier2, ...] noua_cale
 - copiază fișierele date ca parametru la calea `noua_cale` (ultimul parametru)

Precizări suplimentare

Căi

- La pornirea programului, utilizatorul se află în root (/), care este inițial un director gol ce reprezintă rădăcina întregului sistem virtual de fișiere.
- Numele de fișiere/directoare conțin doar litere, cifre și caracterul punct (.). Nu se vor folosi ghilimele sau apostrofuri în comenzi.
- Directorul '.' reprezintă directorul curent, iar directorul '..' reprezintă părintele directorului curent.
- Orice cale poate fi atât absolută (începe cu '/'), cât și relativă (poate - sau nu - să înceapă cu './').
- O cale poate avea (doar) la sfârșit caracterul special de wildcard - '*', reprezentând toți fiii (din arborele de prefixe) nodului ce corespunde șirului de caractere din fața wildcard-ului.

Erori

- Dacă o cale nu este găsită (în cazul comenzilor `cd`, `rm`, `mv`, `cp`), se va afișa la mesajul `"PATH: No such file or directory"`, unde `PATH` este calea căutată.
- Dacă un fișier/director există deja (în cazul comenzilor `mkdir`, `touch`), se va afișa mesajul `"PATH: already exists"`, unde `PATH` este fișierul căutat.
- Dacă în locul unui director se dă calea unui fișier (în cazul comenzilor `cd`, `rmdir`), se va afișa mesajul `"PATH: Not a directory"`, unde `PATH` este directorul căutat.
- Dacă directorul ce se vrea a fi șters (în cazul comenzii `rmdir`) nu este gol, se va afișa mesajul `"PATH: Directory not empty"`, unde `PATH` este directorul căutat.
- Dacă o comandă este apelată cu prea puțini parametri, se va afișa mesajul `"CMD: missing operand"`, unde `CMD` este comanda respectivă.
- Dacă o comandă este apelată cu prea mulți parametri, se va afișa mesajul `"CMD: too many arguments"`, unde `CMD` este comanda respectivă.
- Dacă o comandă invalidă este apelată, se va afișa mesajul `"CMD: command not found"`, unde `CMD` este comanda respectivă.

Exemple de rulare

În următoarele exemple, considerăm că ne aflăm în rădăcina sistemului prezentat mai sus. Inputul utilizatorului este reprezentat cu culoarea verde, iar output-ul dat de program cu negru:

1. Pentru a muta fișierul `a.out` în calea `/ana/are/mere/` vor fi executate următoarele comenzi:

```
ls
abc ana a.out man manual
mv a.out ana/are/mere/
ls -F
abc/ ana/ man* manual/
ls -F abc
a/ b/ c/
ls -F abc/a

ls -F abc/b

ls -F abc/c

ls -F ana
are/
ls -F ana/are
mere/
ls -F ana/are/mere
a.out*
ls -F manual
page1* page2* page3*
```

2. Accesare director `/abc/b/`

```
ls
abc ana a.out man manual
pwd
/
cd /abc/b
pwd
/abc/b
```

3. Navigare sistem de fișiere și afișarea conținutului directoarelor

```
cd manual/
ls
page1 page2 page3
cd ..
```

pwd

/

ls

abc ana a.out man manual

ls abc

a b c

ls ana

are

ls ana/are

mere

ls manual

page1 page2 page3

4. Comenzi greșite

mkdir

mkdir: missing operand

rmdir abc

./abc: Directory not empty

touch a.out

a.out: already exists

makedir

makedir: command not found

Notare

- **85 de puncte:** conform testelor de pe **vmchecker**
- **10 puncte:** coding style; O temă care primește maxim pe coding style nu trebuie să conțină:
 - Linii de cod mai lungi de 80 de caractere
 - Warning-uri de compilare (tema trebuie compilată cu flag-ul -Wall)
 - Folosirea funcțiilor de alocare de memorie (malloc/calloc) fără a verifica faptul că acestea au reușit să realizeze alocarea
 - Folosirea greșită a pointerilor
 - Nume de variabile și funcții nesugestive
 - Cod ilizibil
- **5 puncte:** README
- **20 de puncte:** pentru implementările ce nu conțin memory leak-uri

Temele care nu compilează sau nu rulează se vor nota automat cu 0 puncte.

Implementările care nu respectă abordarea de trie sau restricțiile impuse vor fi depunctate.