

UNIVERSITATEA POLITEHNICA BUCUREȘTI
FACULTATEA ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

SISTEME DE FIȘIERE ÎN LINUX
SISTEME DE FIȘIERE DISTRIBUITE

CIUREA CĂTĂLIN

Master IISC

CUPRINS

Introducere.....	3
1.Notiuni generale si aspecte asupra sistemelor de fisiere.....	4
1.1 Nivelul Virtual File System.....	4
1.2 I-node-ul. Concepte. Descriere	5
2.Tipuri de sisteme de fisiere.....	6
2.1 Sisteme de fisiere pentru discuri. Sisteme cu jurnalizare	6
2.2 Sisteme de fisiere distribuite	7
2.3 Sisteme de fisiere speciale	8
3.Sisteme de fisiere utilizate in prezent in Linux.	
3.1 EXT2.....	8
3.2 EXT3.....	9
3.3 RaiserFS	10
3.3 EXT4.....	11
4.Sisteme distribuite de fisiere.	
4.1 Network File System(NFS).....	12
4.2 Server Message Protocol(SMB).....	14

INTRODUCERE

Sunt peste 30 de ani de când a apărut sistemul de operare UNIX. Multe lucruri s-au schimbat de la lansarea sa, acesta ajungând de la o platformă pentru dezvoltarea de aplicații software la un sistem de operare pentru sisteme workstation sau ca o platformă pentru servere.

Transformarea sistemului de operare UNIX într-un instrument utilizat pentru rularea de aplicații enterprise a fost influențată și de evoluția sistemelor de fișiere care au un rol foarte important într-un sistem de operare. Facilități precum jurnalizarea, clustering-ul sau snapshot-ul au contribuit la creșterea semnificativă a importanței sistemelor de fișiere.

Pe parcursul vieții lor, fișierele sunt supuse multor operații: ele trebuie create, denumite, regăsite, renumite, citite, modificate, duplicate, deplasate, reorganizate, defragmentate, șterse. În tot acest timp este nevoie și de o corespondență clară între numele unui fișier, și adresa sa de pe dispozitivul fizic de stocare. Regăsirea rapidă a unui fișier este și ea importantă, în condițiile în care un singur dispozitiv de stocare poate găzdui chiar și sute de mii de fișiere. Pentru a asigura viteze de funcționare mulțumitoare, organizarea internă a stocării și accesării fișierelor unui hard-disk trebuie să țină cont și să profite de caracteristicile acestuia.

1.NOȚIUNI GENERALE ȘI ASPECTE ASUPRA SISTEMELOR DE FIȘIERE

Noțiunea de sistem de fișiere reprezintă organizarea fișierelor și a datelor conținute de acestea într-un sistem de operare, pentru a facilita căutarea și accesarea lor. Sistemele de fișiere pot utiliza un dispozitiv de stocare precum un hard-disk sau un CD și rolul lor principal este menținerea adreselor fizice a fișierelor.

Ca o definiție mai formală, un sistem de fișiere reprezintă o bază de date specială destinată stocării, organizării și a manipulării datelor.

Cele mai multe sisteme de fișiere se folosesc de un dispozitiv de stocare care oferă acces la o matrice de sectoare fizice de dimensiune fixă (în general 512 bytes sau 1,2,4 KBytes). Soft-ul sistemului de fișiere este responsabil cu organizarea acestor sectoare în fișiere și directoare și menținerea legăturilor între fișiere și sectoarele de care aparțin. Cele mai multe sisteme de fișiere adresează datele în blocuri de un anumit număr de sectoare(1-64). Sistemele de fișiere nu au nevoie în mod obligatoriu de un dispozitiv de stocare și poate fi folosit pentru a organiza și reprezenta accesul la orice fel de date indiferent dacă sunt stocate sau sunt generate dinamic.

Sistemele de operare Linux sunt foarte flexibile și au o lungă istorie de interoperabilitate cu alte sisteme pe diverse platforme hardware. O consecință a acestei compatibilități a sistemelor Linux cu alte sisteme de operare este aceea că sistemele Linux pot citi și scrie pe diferite sisteme de fișiere care au fost create pentru alte sisteme de operare mult diferite de Linux. Un motiv pentru care Linux-ul suportă atât de multe tipuri de sisteme de fișiere este modul de realizare a nivelului VFS (Virtual File System). Nivelul VFS este un nivel de abstractizare dintre kernel și spațiul utilizatorului care execută diverse comenzi.

1.1 Nivelul *Virtual File System*

Nivelul VFS evită duplicarea codului identic între sistemele de fișiere și oferă o metodă universală de acces la suportul fișierelor. Ideea de bază din spatele VFS este în introducerea unui model de fișier capabil să reprezinte toate sistemele de fișiere suportate. Acest model oglindește în mod strict modelul de fișier de bază din sistemul tradițional UNIX. Fiecare implementare de sistem de fișiere trebuie să își translateze structura sa la cea a modelului *Virtual File System*. În modelul de bază al fișierului, fiecare director este privit ca un fișier care conține o listă cu alte fișiere și directoare. Alte sisteme de fișiere care nu își au originile din UNIX folosesc tabela FAT(File Allocation Table) care stochează poziția fiecărui fișier în arborele directorului. În aceste tipuri de sisteme de fișiere, directoarele nu sunt fișiere. Pentru a putea păstra modelul de bază al VFS-ului, implementările din Linux a acestor sisteme de fișiere bazate pe tabele FAT trebuie să fie capabile să construiască atunci când este nevoie, fișierele corespunzătoare directoarelor. Astfel de fișiere există numai ca obiecte în memoria kernelului.

1.2 *I-node-ul. Concepte. Descriere*

Noțiunea fundamentală a multor sisteme de fișiere UNIX este *i-node*-ul. Fiecărui fișier din sistem îi este asociat un astfel de *i-node*. *I-node*-ul este o structură de control care poartă către blocuri de date sau către alte *i-node*-uri. Informația de control conținută într-un *i-node* este alcătuită din: permisiuni, proprietarul fișierului, dimensiune, data ultimei accesări, data creării. Un lucru care nu este menținut într-un *i-node* este numele fișierului deoarece, având în vedere că un fișier are asociat un singur *i-node* ar fi un spațiu irosit memorarea fiecărui nume de fișier în *i-node*-uri.

Directoarele sunt și ele instanțe speciale de fișiere și deci fiecare director are și el asociată o structură de date de tip *i-node* care va referi blocuri de date care conțin informații despre fișierele din director. Pe multe tipuri de sisteme de fișiere, numărul disponibil de *i-node*-uri este limitat la inițializarea sistemului de fișiere și în consecință și numărul total de fișiere care pot fi scrise pe hard-disk este limitat. O fracțiune tipică a spațiului alocat pentru *i-node*-uri este 1% din dimensiunea discului. Totuși unele sisteme de fișiere moderne precum ext4 sau XFS suportă extindere sau alocare dinamică de *i-node*-uri, facilități care permit mărirea spațiului sistemului de fișiere sau/și a numărului de *i-node*-uri.

Numărul *i-node*-ului indexează un tabel de *i-node*-uri într-o locație cunoscută. Utilizând numărului *i-node*-ului, kernelul poate accesa conținutul acestuia, incluzând referințele la datele propriu-zise ale fișierului.

Caracteristicile principale ale unui *i-node* sunt așadar:

- nu conțin nume de fișiere, ci numai metadate.
- directoarele sunt liste de structuri, fiecare conținând un nume de director și un număr de *i-node*
- kernelul caută un director după numele său și apoi face corespondența dintre numele directorului și numărul *i-node*-ului.

Structura unui *i-node*:

- lungimea fișierului în bytes
- *device ID*(identifică dispozitivul pe care este stocat fișierul)
- *user ID*(identifică proprietarul fișierului)
- *group ID*(identifică grupul din care face proprietarul fișierului)
- *file mode*(tipul fișierului și permisiunile de acces la fișier)
- *flag-uri* adiționale care limitează utilizarea și modificarea fișierului
- *timestamps*(data ultimei modificări, data ultimei accesări)
- *link count*(contor de referințe care poartă spre *i-node*-ul în cauză)
- *pointeri* către blocurile de date de pe hard-disk

Unui nume de fișier îi corespunde un singur *i-node* dar la un *i-node* pot fi legate mai multe nume de fișiere. Sistemele de fișiere cu *i-node-uri* au proprietăți care le diferențiază față de alte sisteme de fișiere.

- Numărul *i-node-ului* va rămâne același pentru un fișier care este mutat pe același disc dar la altă adresă. Această proprietate nu poate fi implementată de sistemele de fișiere FAT care nu au nici o metodă de a păstra proprietățile unui fișier atunci când atât acesta cât și directorul care îl conține sunt mutate la altă adresă.
- Instalarea librărilor este simplă atunci când se utilizează sisteme de fișiere cu *i-node-uri*. Un proces poate avea o librărie asociată cu un fișier în timp ce alt proces poate schimba fișierul prin crearea unui nou *i-node* și în felul acesta noile librării sunt mapate fără a fi necesară restartarea sistemului.
- Posibilitatea asocierii mai multor *hard-link-uri* către același *i-node*. Un *hard-link* reprezintă o referință la un fișier iar atunci când un fișier are mai multe *hard-link-uri* asociate, atunci orice modificare este “văzută” de toate referințele. Legăturile simbolice, prin contrast cu legăturile *hard*, nu sunt referințe la fișierele respective ci la numele acestora.

2.TIPURI DE SISTEME DE FIȘIERE

Sistemele de fișiere pot fi clasificate în: sisteme de fișiere pentru discuri, sisteme de fișiere distribuite și sisteme de fișiere speciale.

2.1 Sisteme de fișiere pentru discuri. Sistemele cu jurnalizare

Un sistem de fișiere pentru disc este proiectat pentru stocarea fișierelor pe dispozitive de memorie care pot fi conectate direct sau indirect la sistemul de calcul. Exemple de astfel de sisteme de fișiere includ sistemele de fișiere FAT(FAT12, FAT16, FAT32, exFAT), NTFS, HFS, HPFS, EXT2, EXT3, VeritasFS. Unele dintre acestea suportă facilitatea de jurnalizare care reprezintă facilitatea de a păstra un istoric al modificărilor efectuate, istoric actualizat înainte de operarea modificărilor pe sistemul de fișiere. Această tehnică reduce cu mult probabilitatea de corupere a sistemului de fișiere în cazul unei erori de sistem sau a întreruperii alimentării.

De exemplu ștergerea unui fișier într-un sistem de fișiere UNIX implică 2 pași:

-ștergerea înregistrării din cadrul directorului

-marcarea spațiului fișierului și *i-node-ului* său ca fiind liberi în tabela de spațiu liber

Dacă sistemul cade între pașii menționați mai sus, va rămâne un *i-node* orfan și astfel o pierdere de spațiu de stocare. Într-un sistem de fișiere fără jurnalizare, detectarea și recuperarea din astfel de inconsistențe necesită o parcurgere completă a

structurilor de date. Aceasta poate dura mult timp dacă sistemul de fișiere este mare sau dacă accesul la el se face cu o lățime de bandă redusă.

Un sistem de fișiere cu jurnalizare păstrează deci un jurnal cu modificările pe care intenționează să le efectueze, înainte de efectuarea lor. După o cădere, recuperarea implică doar reefectuarea modificărilor aflate în jurnal până când sistemul redevine consistent. Modificările sunt astfel atomice (sau indivizibile) prin aceea că fie au succes (au avut succes dinainte de cădere sau au fost reefectuate cu succes la recuperare), fie nu sunt efectuate deloc (sunt sărite deoarece nu au apucat să fie scrise complet în jurnal).

Unele sisteme de fișiere permit creșterea, reducerea dimensiunii și realocarea jurnalului ca și în cazul unui fișier normal, dar majoritatea își păstrează jurnalul într-o zonă contiguă sau într-un fișier ascuns despre care se știe că nu își va schimba locul sau dimensiunea la montarea sistemului de fișiere.

Un jurnal fizic păstrează copii ale blocurilor ce urmează a fi scrise așa cum este cazul lui Ext3fs. Un jurnal logic păstrează doar informații despre modificări într-un format compact special, cum este cazul XFS și NTFS. Aceasta reduce cantitatea de date ce trebuie citită și scrisă din și în jurnal dar folosește operații mari cu metadate.

Alte implementări de sisteme de fișiere evită jurnalizarea și implementează în schimb actualizări soft: comandă scrierile în așa fel încât sistemul de fișiere de pe disc nu este niciodată inconsistent, sau așa încât singura inconsistență ce poate fi creată în caz de cădere este o pierdere de spațiu de stocare. Pentru a recupera aceste pierderi, tabela de spațiu liber este reactualizată în urma unei verificări complete a sistemului de fișiere în cauză.

Jurnalizarea poate avea un impact puternic asupra performanței, deoarece necesită ca toate datele să fie scrise de două ori. Jurnalizarea cu metadate este un compromis între siguranță și performanță. În acest caz se stochează în jurnal doar niște metadate. Aceasta asigură că sistemul poate fi recuperat rapid la următoarea montare, dar coruperea datelor poate avea loc deoarece datele nejournalizate și metadatele din jurnal pot să ajungă să fie desincronizate.

2.2 Sisteme de fișiere distribuite.

Cea de-a doua categorie de sisteme de fișiere este reprezentată de sistemele de fișiere distribuite. Un sistem de fișiere distribuit permite accesul la fișiere care aparțin mai multor sisteme de calcul, printr-o rețea de calculatoare. Este deci posibil ca mai multi utilizatori să împartă fișiere și alte resurse.

Sistemele de calcul din rețea nu au acces direct la blocurile de date dar comunică prin rețea prin intermediul unui protocol. Acesta oferă posibilitatea restricționării accesului la sistemul de fișiere prin liste de acces. În contrast cu sistemele de fișiere distribuite, sistemele de fișiere împărțite (*shared*) oferă acces tuturor nodurilor la blocurile de date. În aceste sisteme, restricționarea accesului trebuie realizată pe sistemul de calcul al clientului.

Sistemele de fișiere distribuite pot include facilități de replicare transparentă și toleranță la erori. Prin acestea înțelegem faptul că atunci când un număr de sisteme de calcul din rețea devin indisponibile, sistemul continuă să funcționeze fără pierderi de date.

Transparența este implementată de obicei în sistemele de fișiere distribuite, astfel încât fișierele accesate prin rețea să poată fi tratate precum fișierele locale.

Cele mai cunoscute sisteme de fișiere distribuite sunt: Network File System(NFS) dezvoltat de Sun Microsystems, Andrew File System(AFS), Apple Filing Protocol(AFP), Server Message Block(SMB), Netware Core Protocol(NCP), 9P.

2.3 Sisteme de fișiere speciale.

Un sistem este sistem de fișier special dacă nu este nici sistem de fișier pentru discuri și nici distribuit. Acestea sunt sisteme la care fișierele sunt aranjate dinamic de softuri speciale. Cel mai adesea, astfel de sisteme de fișiere sunt utilizate pe sistemele de operare cu structuri bazate pe fișiere cum sunt sistemele UNIX. Exemple de astfel de sisteme de fișiere sunt: UnionFS, AUFS, Cascade File System, WebDAV.

Sistemul de fișiere UnionFS este un sistem care combină în mod transparent conținutul mai multor sisteme de fișiere, menținând însă conținutul fizic al fiecăruia separat. Atunci când se face această suprapunere a directoarelor la nivel logic, există priorități astfel încât dacă există fișiere cu același nume în directoare separate, unul va avea prioritate în fața celuilalt.

3.SISTEME DE FIȘIERE UTILIZATE ÎN PREZENT ÎN LINUX.

3.1 EXT2

Sistemul de fișiere EXT2 a fost dezvoltat pentru kernelul Linux-ului ca un înlocuitor pentru predecesorul său EXT. EXT2 a fost pentru o vreme sistemul de fișiere predefinit pentru multe distribuții Linux printre care și Debian și Red Hat pînă când a fost înlocuit la rîndul său cu sistemul de fișiere EXT3. În prezent se mai folosește sistemul EXT2 pentru memoriile de tip flash cum sunt cardurile SD, SSD sau memoriile USB, deoarece nu are facilitatea de jurnalizare, care mărește numărul de scrieri pe dispozitivul de memorare de tip flash, care are un număr limitat de cicluri de scriere.

Spațiul este organizat în blocuri grupate pentru a minimiza timpul de acces la disc atunci când se citesc informații succesive. Superblocul, care este o informație esențială pentru pornirea sistemului de operare este copiat și stocat la fiecare 8192 de blocuri.

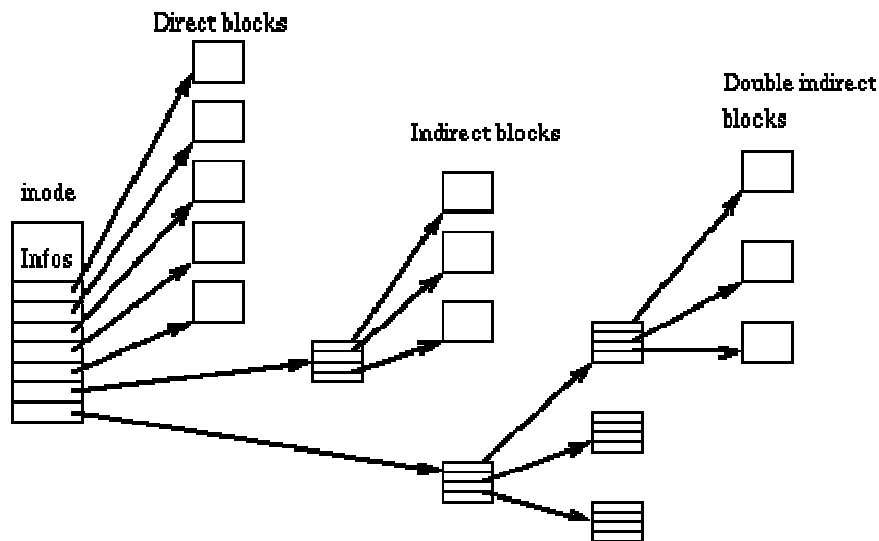


Fig.1 Exemplu de organizare a i-node-urilor la sistemul EXT2

EXT2 a fost implementat astfel încât să fie ușor de adăugat noi facilități. Este flexibil, suportă fișiere de până la 4GB cu nume de până la 1012 caractere. EXT2 rezervă în general 5% din blocurile de pe disc pentru a fi folosite de către utilizatorul root pentru a rezolva situația în care utilizatorul încarcă sistemul de fișiere la maximum. Numărul maxim de subdirectoare este de 32768, iar dacă se depășește numărul de 10000 sau 15000 de fișiere într-un director, utilizatorul este atenționat de faptul că operațiile pot dura foarte mult timp. Dischetele de bootare Red Hat folosesc sistemul de fișiere de tip EXT2.

3.2 EXT3

Sistemul de fișiere EXT3 a fost introdus în anul 2001 în kernelul Linux și scopul său principal a fost să rezolve problema timpului îndelungat necesar efectuării unei verificări complete după o cădere a sistemului. Patru sisteme de fișiere au fost realizate ca opțiuni pentru înlocuirea învechitului EXT2. Acestea erau: EXT3, RaiserFS, XFS, JFS. Toate ofereau rezolvarea problemei mai sus menționate prin facilitatea de jurnalizare care a fost descrisă în capitolul anterior.

Chiar dacă performanțele lui EXT3 sunt mai scăzute decât a altor sisteme de fișiere precum RaiserFS, XFS sau JFS, are un avantaj major în faptul că acesta are nevoie de volum mai mic de calcul al procesorului și de asemenea este considerat a fi mai sigur datorită simplității sale și a testării sale mai intense. Deși lipsa unor facilități moderne precum alocarea dinamică de *i-node-uri* ar putea fi considerată un dezavantaj, în situația de recuperare a datelor, EXT3 are avantaj în fața sistemelor cu această

facilitate deoarece metadatele se află în locații fixe bine știute și există o anumită redundanță în structurile de date care poate permite sistemului de fișiere EXT3 să recupereze datele corupte, lucru care în general nu poate fi realizat de către sistemele de fișiere de tip arbore.

Limita unei partiții EXT3 este de 32 TB în condițiile în care dimensiunea unui bloc este de 8 KB. Dimensiunea unui fișier este limitată la 2 TB.

Există 3 nivele de jurnalizare implementate în EXT3:

- *Journal*(risc minim) – atât metadatele cât și conținutul fișierelor sunt scrise în jurnal înainte de a fi scrise pe disc. Performanța este redusă deoarece datele sunt scrise practic de 2 ori. O dată în jurnal și o dată pe sistemul de fișiere propriu-zis.
- *Ordered*(risc mediu) – doar metadatele sunt scrise în jurnal. Este garantat faptul că datele sunt scrise pe disc înainte ca metadatele să fie marcate ca fiind scrise în jurnal. Acest mod este predefinit în multe distribuții Linux.
- *Writeback*(riscul cel mai mare) – doar metadatele sunt scrise în jurnal și pot fi scrise înainte sau după ce jurnalul este actualizat. Fișierele modificate exact înaintea unei căderi a sistemului, pot deveni corupte.

Dezavantajele lui EXT3:

- Deoarece s-a intenționat să fie compatibil cu EXT2, nu are implementate facilități moderne precum alocarea dinamică de *i-node-uri*.
- Nu există nici o facilitate de defragmentare a volumului la nivel de sistem de fișiere.
- Nu există suport pentru recuperarea fișierelor șterse. EXT3 șterge *i-node-ul* la ștergerea unui anumit director, iar ștergerea este cel mai probabil permanentă.
- Nu face verificarea jurnalului(*checksumming*)

3.3 RaiserFS

Este un sistem de fișiere cu jurnalizare destinat serverelor unde sunt necesare performanțe sporite. Este mai eficient din punct de vedere al spațiului ocupat deoarece nu ocupă minimul de 1 bloc pentru fiecare fișier. Dacă sunt scrise mai multe fișiere foarte mici pe disc, acestea vor fi unite într-un singur bloc. De asemenea RaiserFS nu are o dimensiune fixă pentru *i-node-uri* ceea ce economisește aproximativ 6% din spațiul discului.

RaiserFS este indicat în aplicațiile în care se lucrează cu fișiere mici cum ar fi sisteme de distribuire a mailurilor, cache-uri HTTP.

Dezavantaje:

- Unele operații cu directoare nu sunt sincronizate ceea ce poate duce la corupere de date
- Nu există programe pentru defragmentarea sistemului de fișiere RaiserFS.
- Primele versiuni de RaiserFS erau puternic instabile și nu erau recomandate pentru lucrul cu Network File System.

Dimensiunea maximă a unui fișier în RaiserFS este de 8 TB, în timp ce dimensiunea maximă a unei partiții este de 16 TB. Numărul maxim de fișiere pe un volum RaiserFS este de 4 miliarde.

3.4 EXT4

EXT4 a fost dezvoltat pentru a aduce îmbunătățiri lui EXT3 dar să mențină totuși compatibilitatea cu acesta. Dimensiunea maximă a unei partiții EXT4 a fost mărită la 1 EB iar dimensiunea maximă a unui fișier este de 16 TB.

Facilități:

- *Extents* – Un *extent* este o zonă continuă de blocuri fizice, utilizată pentru a spori performanța la utilizarea fișierelor mari. Un singur extent poate conține 128 MB spațiu continuu cu dimensiunea blocului de 4 KB. Într-un *i-node* pot fi stocate 4 *extent-uri*.
- Prealocarea de spațiu – EXT4 oferă facilitatea de prealocare a spațiului pe disc pentru un fișier. Metoda clasică presupunea scrierea spațiului fișierului cu biți de 0 pentru a rezerva spațiul atunci când fișierul este creat. EXT4 folosește un apel de sistem care a fost introdus în kernelul Linuxului. Spațiul alocat pentru astfel de fișiere este garantat și cel mai probabil este continuu. Această facilitate are aplicabilitate în streaming și baze de date.
- EXT4 a urcat limita numărului de subdirectoare la 64000. Pentru a spori performanțele în cazul lucrului cu directoare foarte mari, se folosește o tabelă de indexare.
- EXT4 a introdus de asemenea facilitatea de verificare a jurnalului (*checksumming*).
- Reduce fragmentarea prin alocarea mai multor blocuri într-o singură operație și încercând totodată să aleagă blocuri care sunt succesive.
- Reduce timpul de verificare a sistemului de fișiere prin marcarea blocurilor nealocate și a secțiunilor corespunzătoare din tabela de *i-node-uri*.

4.SISTEME DE FIȘIERE DISTRIBUITE.

Anii 1980 au dus la apariția a multor sisteme de fișiere distribuite incluzând *Network File System*(NFS) al celor de la SUN, *Andrew File System* (AFS) care a evoluat în *Distributed File Service*(DFS). Unele sisteme de fișiere au dispărut rapid. De departe cel mai de succes sistem de fișiere distribuit a fost NFS fiind folosit și astăzi pe sistemele Linux.

Sistemele de fișiere distribuite operează în jurul unui model client/server, cu una din mașini având sistemul de fișiere și împărțindu-l clienților printr-un protocol bine definit. Protocolul și modul de transfer al fișierelor către utilizator este transparent. Spre deosebire de sistemele de fișiere locale unde spațiul de memorie este atașat fizic și este accesibil doar de către procesele locale, sistemele de fișiere distribuite permit accesul la fișiere pe o altă mașină printr-un protocol.

4.1 NETWORK FILE SYSTEM(NFS)

Obiectivele NFS au fost:

- Independența de platformă și de sistemul de operare
- Transparența în cazul în care serverul devine indisponibil
- Acces transparent la fișiere
- Performanțe acceptabile

Sunt 3 componente principale ale NFS-ului: protocolul, clientul și serverul.

La început protocolul NFS opera exclusiv prin UDP. Proiectanții și-au dorit ca pe server să nu fie menținute informații despre cererile anterioare. Aceasta evită orice mecanism de recuperare după o cădere a serverului. Dacă, clientul nu primește răspuns după o perioadă de timp, cererea este retrimisă până când va avea succes. Acest comportament asigură faptul că clientul nu va realiza diferența între o cădere a serverului sau un răspuns lent al acestuia.

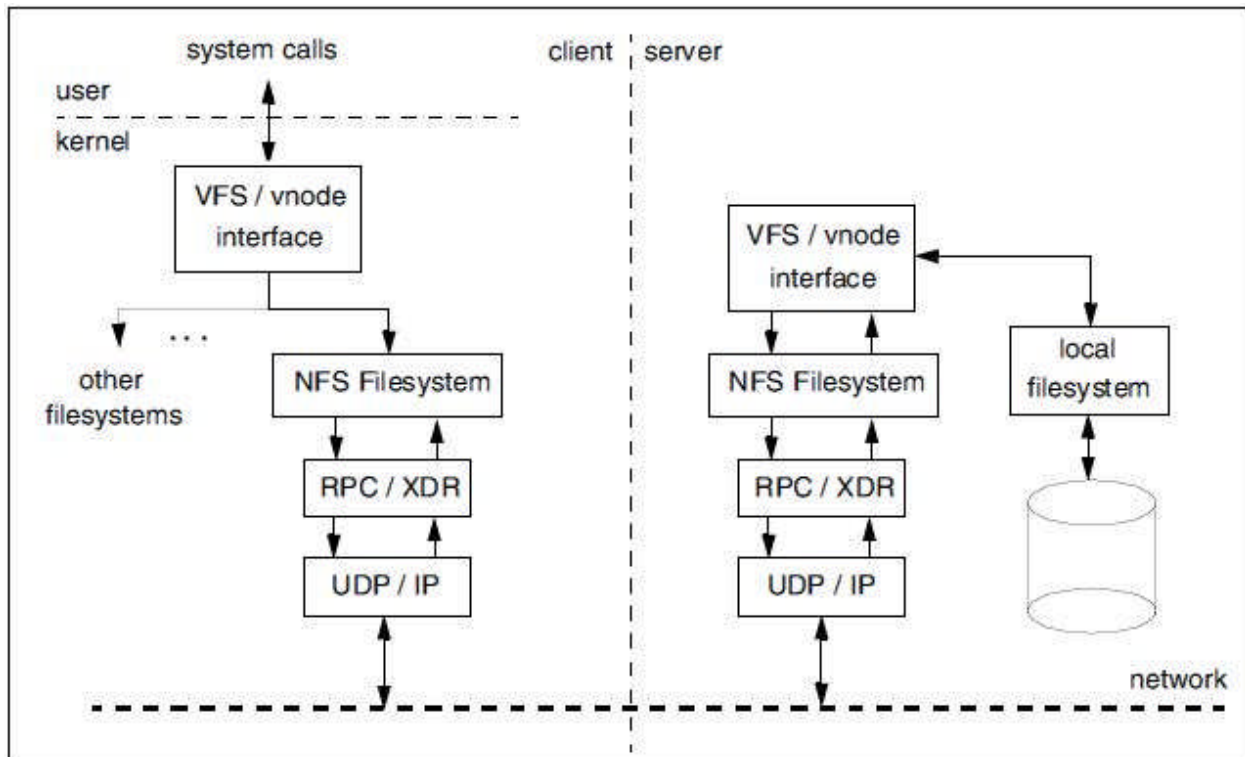


Fig.2 Diagramă de funcționare a NFS

Comunicarea dintre server și client prin NFS se bazează atât pe *Remote Procedure Call*(RPC) cât și pe *External Data Representation*(XDR). XDR descrie modul de codare a tipurilor de date. RPC oferă o infrastructură pentru crearea de aplicații client/server prin care clientul poate apela o funcție oferită de către server la fel cum ar apela o funcție din propriul spațiu de adrese.

În versiunea NFS 3 a fost introdus suportul pentru protocolul TCP pentru a evita pierderea pachetelor în rețea

. Versiunea 2 a protocolului avea 2 mari probleme: limitarea accesului la fișiere de maxim 4 GB și obligativitatea scrierilor sincronizate pe server.

NFS 3 a rezolvat pe lângă aceste două probleme și următoarele:

- Versiunea 2 a protocolului limita dimensiunea fișierelor și a căilor de acces la 255 și respectiv 1024 caractere. În versiunea 3 acestea au fost înlocuite cu șiruri de caractere variabile a căror lungime putea fi stabilită între client și server.
- A restrâns numărul de erori care putea fi returnat de server. Valorile erorilor sunt iterate și nici o valoare din afara listei nu este permisă.
- A fost scoasă restricția de a scrie maxim 8 KB la un apel.

Protocolul NFS a fost dezvoltat pentru rețele LAN și a fost lansat înainte de răspândirea internetului. Versiunea 4 a adus modificări astfel încât protocolul să poată

fi utilizat în rețele WAN și să ofere totuși un nivel de securitate acceptabil. Cea mai importantă modificare a versiunii 4 este faptul că serverul menține informații despre c

Unele din facilitățile aduse de versiunea 4 așadar sunt: rețerile anterioare ale clienților.

- Codarea numelor de fișiere și directoare cu UTF-8. Astfel s-a oferit suport pentru internaționalizare. Înainte era folosit ASCII, și puteau fi situații în care se foloseau caractere mixte, iar XDR nu putea specifica modul de codare.
- *File-locking*- oferă o siguranță mai mare prin adăugarea de funcții de blocare la nivel de fișier.
- *Cache-ul la nivelul clientului*

4.2 SERVER MESSAGE Block(SMB)

SMB, operează la nivelul aplicație în stiva de protocoale și este utilizat în principal pentru a oferi acces la fișiere partajate, imprimante sau la porturi seriale. Oferă de asemenea un mecanism de autentificare. SMB este utilizat îndeosebi pe sistemele Windows și mai este cunoscut și sub denumirea de Microsoft Windows Network.

În cadrul protocolului SMB trebuie să avem în discuție următoarele:

- Specificațiile protocolului SMB
- Serviciile de server și workstation care implementează protocolul în Windows.
- Serviciul Samba care implementează protocolul în sistemele Unix.
- Serviciile care utilizează SMB ca un canal de autentificare inter-proces

SMB-ul utilizează protocolul TCP la nivelul transport și comunică prin portul 445.

SMB funcționează printr-o arhitectură client/server, unde clientul efectuează o cerere iar serverul răspunde în consecință.

Proiectanții de rețele au realizat faptul că latența are un impact major asupra performanței SMB 1.0 care avea rezultate mai slabe decât FTP. Acest lucru se datorează faptului că SMB lucra la nivel de bloc și a fost proiectat de la bun început pentru rețele LAN de mici dimensiuni. Limita blocului de date este de 64 Kb.

Microsoft a lansat versiunea SMB2 împreună cu sistemul de operare Vista în 2006. Chiar dacă protocolul este proprietar, specificațiile au fost publicate pentru a permite altor sisteme să poată interopera cu sistemele de operare Windows.

- A fost adăugată abilitatea de a adăuga mai multe acțiuni într-o singură cerere, reducând numărul de apeluri client-server.
- A fost mărită dimensiunea bufferului pentru a spori viteza de transfer a fișierelor de dimensiuni mari.
- SMB1 utiliza date pe 16 biți care limita dimensiunea unui bloc la 64 KB. Versiunea 2 utilizează date pe 32,64 sau chiar 128 de biți eliminând practic limitarea.

SMB2 aduce două mari beneficii pentru Windows.

- Drepturile de proprietate intelectuală

- Reducere a complexității protocolului în ideea de a fi compatibil în principal cu sistemele Windows.

BIBLIOGRAFIE

- 1.UNIX Filesystems – Evolution, Design and Implementation, 2003, **Steve Pate**
- 2.Red Hat Linux Networking and System Administration, 2002, **Terry Collings, Kurt Wall**
- 3.Red Hat Linux Administration.A Beginner's Guide, 2003, **Michael Turner, Steve Shah**
- 4.Linux Bible 2006, **Christopher Negus**
- 5.Operating System Concepts, 2005, **Galvin Silberschatz**
- 6.en.wikipedia.org
- 7.ro.wikipedia.org