CHAPTER 11

Implementing File-System

(69 Questions)

01: What is ***File-system Structure***?

02: What are the ***functions of File System***? What does disk provide?

03: What is ***File Control Block***?

04: What does ***Device Driver*** do?

05: What are the ***Layers*** that ***organize File System***?

06: What is the ***given command for Basic File System***?

07: What does Basic ***File System manage in Memory***?

08: What is a ***File Organization Module***?

09: What is the ***Logical File System***?

10: What is the ***benefit for layering the file system***?

11: What are ***the structures in a memory*** that are ***used to implement a file system***?

12: What does ***Boot Control Block*** contain?

13: What does ***Volume Control Block*** contain?

14: What does ***File Control Block*** (FCB) contain?

15: What is a ***Typical File Control Block***?

16: What is a ***Partition***?

17: What does ***Root Partition*** contain?

18: Is all metadata correct?

19: What do **Virtual File Systems (VFS)** on Unix Provide?

20: What is the function of ***Virtual File Systems***?

21: What does ***VFS define***?

22: Mention the two ***methods of implementing a directory***?

23: What are ***Collisions***?

24: What are the major methods of allocating disk space are in wide use?

25: What is ***Contiguous Allocation***? What is its ***features***?

26: What it needs for problems include finding space for file, knowing file size, external fragmentation?

27: What does many newer file systems use?

28: What do ***Extent-based file systems allocate***?

29: What is ***Linked Allocation***?

30: What are ***features of Linked Allocations***?

31: what is an important variation on linked allocation?

32: What is ***Indexed allocation***?

33: What are the ***mechanisms for Index Block*** purpose?

34: What is the ***Performance of Allocation Methods***?

35: Why do we consider that ***indexed more complex***?

36: What does ***File System maintain***?

37: What are the Approaches to Free space Management?

38: What are the functions of Space Maps?

39: What are a variety of techniques ***Efficiency and Performance*** of Secondary Storage to improve?

40: What is ***Buffer Cache***?

41: What is the difference between ***Synchronous and Asynchronous***?

42: What is ***Page Cache***?

43: What is ***Unified Buffer Cache***?

44: What is ***Recovery***?

45: What is ***Consistency Checking***?

46: A **Recovery** is using system programs for?

47: How to ***lose the Recover*** its file or disk?

48: What is ***Log Structured*** (or ***journaling***)?

49: What are the attributes of **Transactions**?

50: What are the ***Transactions in the Log***? When will it be ***removed from the Log***?

51: What is The ***Network File System (NFS)***?

52: What is the ***Implementation of NFS***?

53: What is ***Remote Directory***?

54: The ***NFS is designed to***?

55: What does ***NFS Specification distinguish***?

56: What is the ***function of Mount Protocol***?

57: What does ***Mount Operation include***?

58: What is ***Export List***?

59: What is ***File Handle***?

60: What does the ***Mount Operation change***?

61: What does ***NFS Protocol provide***?

62: Mention the ***Operations that supported by Remote Procedures***?

63: What are NFS Servers?

64: Does the NFS Protocol provide concurrency-control mechanism?

65: What are the three major Layers of NFS Architecture?

66: What is the performance of NFS Path-Name Translation?

67: How to make lookup faster?

68: When do File-blocks cache?

69: When the cached file blocks are used?

End of Questions (Chapter 11).

Chapter 11

Implementing File-System

1: **File system Structure:** هي وحدة تخزين منطقية وتجميع لمعلومات ذات صلة.

2: **File system:** نظام الملف يقيم في وحدة التخزين الثانوي (disks):

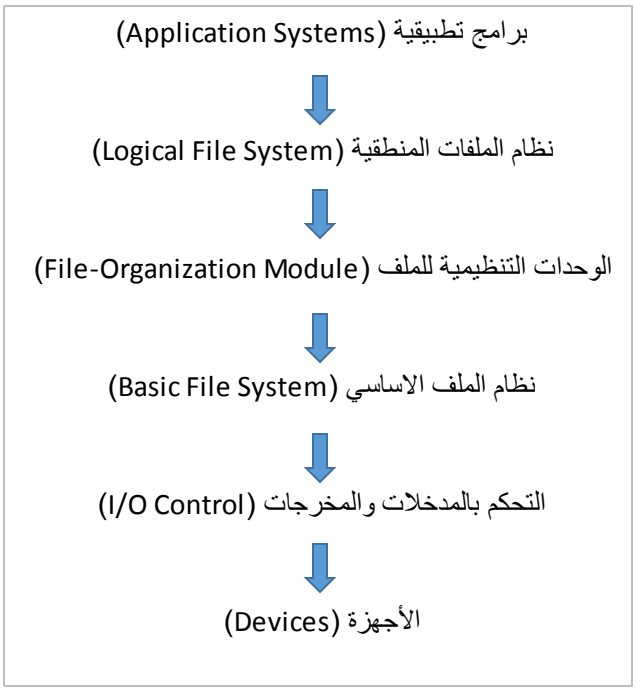
i. يقوم بتوفير واجهة مستخدم لوحدة التخزين، ورسم تخطيطي منطقي فيزيائي.
ii. يوفر الكفاءة والوصول المتلائم إلى القرص وذلك بالسماح بتخزين البيانات ووضعها في موقع يسهل استرجاعها.

- Disk يقوم بتوفير مكان لاعادة الكتابة والوصول العشوائي:
  - I/O يقوم بالتحويلات التي اجريت في الـ Blocks و Sectors.

3: **File Control Block:** هيكلة تخزينية تتألف من معلومات عن الملف.

4: **Device Driver:** يتحكم بالجهاز الفيزيائي. ويدير اجهزة O/I في طبقة التحكم بالمدخلات والمخرجات.

5: نظام الملف (File System) تكون منظمة على شكل طبقات (Layers):



6: **من أمثلة الأوامر المعطاه لنظام الملف الاساسية:**

(read drive1, cylinder 72, track 2, sector 10, into memory location 1060" outputs low level hardware)

7: **نظام الملف الاساسي يدير ايضا ذاكرة buffers و caches (تخصيص، تحرير، تبديل):**

- **Buffers** يحتفظ بالبيانات في transit
- **Caches** يحتفظ بالبيانات المستخدمة بشكل مستمر.

8: **الوحدات التنظيمية للملف (File-Organization Module):** يفهم الملفات، والعنوان المنطقي، و Physical blocks.

9: **نظام الملفات المنطقية (Logical File system) يقوم بإدارة معلومات الـ metadata كالآتي:**

a) يترجم اسم الملف إلى رقم الملف أو قيادة الملف (file handle) أو الموقع وذلك بصيانة File control blocks.
b) دليل الادارة.
c) الحماية.

10: نظام الملفات الطبقية مفيدة للحد من التعقيد والتكرار، ولكنها تضيف overhead وتستطيع أن تقلل من الاداء.

فالطبقات منطقيا بإمكانها أن تكون منفذة بأي طريقة باستخدام الكود بناءا على مصمم نظام التشغيل.

11: تنفذ ملف النظام وظائفها عن طريق الـ Disk وايضا عن طريق Memory structure.

12: **Book Control Block:** يحتوي على معلومات مطلوبة بواسطة النظام لتمهيد نظام التشغيل من هذه Volume.

13: **Volume Control Block:** ويطلق عليه ايضا (superblock, master file table) ويحتوي على تفاصيل الـ volume.

14: **كل ملف من File Control Block (FCB) يحتوي على العديد من التفاصيل عن الملف:**

- Inode أعداد وأذونات وحجم وتواريخ.
- NFTS يخزن في جدول master file باستخدام علاقة هياكل DB.

15: A typical file-control block:

(1) **file permissions** (2) **file dates "create, access, write"** (3) **file owner, group, ACL**

(4) **file size** (5) **file data blocks or pointers to file data blocks**

16: الأقسام (Partitions): عندما يجزأ القرص إلى عدة أقسام فإن كل قسم يسمى partition. وهذا الـ partition إما يحتوي على ملف نظام "cooked" أو لا يحتوي على ملف نظام "raw" وهي عبارة عن سلسلة من ال blocks لا تحتوي على ملف نظام.

Boot Block: يشير إلى Boot Volume أو Boot leader مجموعة من ال blocks التي تحتوي على أكواد كافية لمعرفة كيفية تخزين النواة من نظام الملف.

17: **Root Partition** هو قسم يحتوي على نظام التشغيل وأما الأقسام الأخرى بإمكانها ان تتمسك بأنظمة التشغيل الأخرى أو بأنظمة الملفات الأخرى أو أن تكون raw:

- تكون Mounted عند boot time.
- وبقية الأقسام (Partitions) يمكن ان تكون mount أوتوماتيكيا أو يدويا.

18: **جميع الـ Metadata:**

- لو كانت غير صحيحة (fix it, try again)
- لو كانت صحيحة (add to mount table, allow access)

19: **أنظمة الملفات الافتراضية (VFS) في اليونيكس** تقوم بتوفير طريقة object-oriented لتنظيم أنظمة الملفات. حيث يسمح باستدعاء نفس نظام واجهة المستخدم (API) ليتم استخدامه للعديد من أنواع أنظمة الملفات.

20: **من أبرز وظائف أنظمة الملفات الافتراضية (VFS):**

1. يقوم بفصل عمليات ملف النظام العام من التفاصيل المنفذة.
2. التنفيذ قد يكون نوع من أنواع ملفات النظام أو نظام ملف الشبكة.
3. ثم يقوم بإرسال العملية إلى نظام الملفات المناسبة التنفيذية الروتينية.

21: **أنظمة الملفات الافتراضية (VFS)** تقوم بتعريف مجموعة من العمليات على الـ Objects التي يجب أن تكون منفذة، فلكل object مؤشر لجدول وظيفي. وكل جدول وظيفي لديه عناوين من الروتينيات لتنفيذ تلك الوظائف على ذلك الـ object.

22: **تنفيذ الدليل (Directory Implementation) لها عدة طرق للاستخدام ومن أهمها:**

A. القائمة الطولية (Linear List): أسهل طريقة في تنفيذ الدليل هو استخدام القائمة الطولية لأسماء الملفات مع المؤشرات في بيانات الـ blocks. وافتراضية الوقت لتنفيذ: وقت البحث الطولي والاحتفاظ بالحروف الابجدية المطلوبة عبر ربطها مع القائمة أو استخدام B+ tree.
B. جدول Hash (Hash Table): هو هيكلة من البيانات المستخدمة لدليل الملف. فالقائمة الطولية تقوم بتخزين مدخلات الدليل ولكن هيكلة بيانات الـ hash تكون أيضا مستخدمة. جدول Hash يقوم بالتقليل من وقت بحث الدليل.

23: **Collisions:** هي حالات عندما يكون هناك اسمين لملفين ممزوجين (Hash) في نفس الموقع.

24: **هناك ثلاث طرق أو أساليب اساسية (Three major methods of allocating) في تخصيص مساحة القرص، وهم:**

1) التخصيص المتجاور (Contiguous Allocation).
2) التخصيص الموصول (Linked Allocation).
3) التخصيص المفهرس (Indexed Allocation).

25: **تخصيص المتجاورة (Contiguous Allocation):** كل ملف يحتل (occupies) مجموعة من الـ blocks المتجاورة.

ومن أبرز خصائصه:

1) أفضل أداء في معظم الحالات.
2) بسيط: يتطلب فقط بداية الموقع (#block) وطول (number of blocks).
3) المشاكل المتضمنة في إيجاد مساحة للملف ومعرفة حجم الملف والتجزئة الخارجية تحتاج إلى compaction off-line or on-line.

26: المشاكل المتضمنة في إيجاد مساحة للملف ومعرفة حجم الملف والتجزئة الخارجية تحتاج إلى Compaction off-line or on-line.

27: العديد من أنظمة الملفات الحديثة تستخدم مخطط التخصيص المتجاور المعدل (modified contiguous allocation scheme).

28: **Extend-based file systems** يقوم بتخصيص disk blocks في نطاقات.

29: **التخصيص الموصول (Linked Allocation):** يقوم بحل كل مشاكل التخصيص المتجاور. مع التخصيص الموصول؛ كل ملف تكون قائمة موصولة لـ disk blocks. والـ blocks قد تكون متناثرة (scattered) في أي مكان على القرص.

30: **من خصائص التخصيص الموصول (Linked Allocation):**

1) الملف ينتهي عند اخر المؤشر.
2) لا توجد تجزئة خارجية.
3) كل block يحتوي على مؤشر للـ block التالي.
4) لا توجد اندماجيات (No Compaction).
5) نظام ادارة المساحة الحرة يستدعى عند الحاجة إلى block جديد.
6) تطوير الكفاءة عن طريق تجميع الـ blocks إلى مجموعات ولكنه يزيد من التجزئة الداخلية.
7) قد تكون هناك مشكلة في الدقة (Reliability).
8) لتحديد موقع الـ Block يمكن ان يأخذ العديد من المدخلات والمخرجات وطلبات القرص.

31: **الاختلاف المهم في التخصيص الموصول هو في استخدام جدول تخصيص الملف (FAT: File Allocation Table)**

ومن أبرز خصائصه:

1) بداية الـ volume له جدول (Table)، ومفهرس بواسطة رقم الـ Block.
2) يشبه إلى حد كبير القائمة الموصولة، ولكنه أسرع على القرص وعلى الـ cacheable.
3) بساطة تخصيص block جديد.

32: **التخصيص المفهرس (Indexed Allocation):** عندما تكون مؤشرات الـ blocks متناثرة او مبعثرة (scattered) مع بعضها البعض فإنه يصعب الوصول إليها ولهذا وجد التخصيص المفهرس ليحل المشكلة وذلك بإحضار جميع المؤشرات مع بعضها البعض في موقع واحد ويسمى index block.

33: **آليات عمل index block تتضمن:**

1) مخطط الوصول (Linked scheme): هو ربط blocks جدول الفهرس ولا يوجد له حجم معين.
2) فهرس متعدد المستويات (Multilevel index).
3) مخطط الدمج (Combined scheme).

34: **اداء طرق التخصيص (Performance of allocation methods):**

1) استخدام الطريقة الافضل تعتمد على نوع وصول الملف مثل المتجاورة تكون قوية في التسلسل والعشوائية.
2) الموصولة تكون جيدة مع التسلسل وليست جيدة مع العشوائي.
3) يظهر نوع الـ access عند الانشاء -> يختار إما المتجاورة او الموصولة.
4) المفهرسة أكثر تعقيدا:
   - Single block access يمكن ان يطلب فهرسين block reads ومن ثم بيانات block read.
   - التجمع (Clustering) بإمكانه المساعدة في تطوير throughput وتخفيض CPU overhead.
   - إمكانية إضافة تعليمات إلى مسار التنفيذ لحفظ disk I/O.

35: **تعتبر التخصيص المفهرس الطريقة الاكثر تعقيدا لعدة اسباب:**

1. Single block access يمكن ان يطلب فهرسين block reads ومن ثم بيانات block read.
2. التجمع (Clustering) بإمكانه المساعدة في تطوير throughput وتخفيض CPU overhead.
3. إمكانية إضافة تعليمات إلى مسار التنفيذ لحفظ disk I/O.

36: **File system** يقوم بصيانة قائمة المساحة الحرة (free-space list) لتتبع blocks/clusters المتاحة.

37: **منهجيات ادارة المساحة الحرة (Free-Space Management):**

1) **ناقلات البت (Bit Vector):** قائمة المساحة الحرة تكون منفذة بشكل متكرر كـ bit map or bit vector.
Bit map يتطلب مساحة اضافية، والسهولة في الحصول على الملفات المتجاورة.
2) **القائمة الموصولة (Linked List):** لا يستطيع الحصول على مساحات متجاورة بسهولة، لا يهدر المساحات، لا يحتاج إلى اجتياز (Traverse) القائمة بالكامل.
3) **التجمع (Grouping):** يقوم بتعديل القائمة الموصولة لتخزين عنوان free blocks التالي في أول free block إضافة إلى المؤشرات للـ block التالي الذي يحتوي على مؤشرات الـ free block.
4) **الاحصاء (Counting):** لأنه في معظم الاحيان تكون المساحات المستخدمة والغير مستخدمة متجاورة، مع تخصيص التخصيص المتجاور، والنطاقات والتجميع (clustering).
   a) يحتفظ بعنوان اول free block ويقوم بإحصاء free blocks المتبع.
   b) لديه قائمة free space وبعد ذلك المدخلات التي تحتوي على العنوانين والاحصائيات.
5) **خرائط المساحة (Space Maps):**
   a) يستخدم في ZFS
   b) يعتبر meat-data I/O على أنظمة ملفات كبيرة جدا (هياكل البيانات الكاملة مثل bit map لا يمكنها ان تكون مناسبة في الذاكرة لأنه يحتوي على العديد من المدخلات والمخرجات).
   c) يقسم مساحة الجهاز إلى وحدات metaslab وادارتها (يعطي volume بإمكانه احتواء المئات من metaslabs).
   d) كل meatslab تكون معرفة مع space map (يستخدم في الحسابات الخوارزمية).
   e) يسجل إلى log file بدلا من نظام الملف (log of all block activity, in time order, in counting format).
   f) نشاط Metaslab يخزن مساحة الخريطة إلى الذاكرة في هيكلة الشجرة الموزونة ومفهرسة عن طريق التعويض:
      - إعادة log إلى تلك الهيكلة.
      - ربط free blocks المتجاورة إلى مدخل واحد.

38: **وظائف خرائط المساحة (Space Maps):**

   g) يستخدم في ZFS
   h) يعتبر meat-data I/O على أنظمة ملفات كبيرة جدا (هياكل البيانات الكاملة مثل bit map لا يمكنها ان تكون مناسبة في الذاكرة لأنه يحتوي على العديد من المدخلات والمخرجات).
   i) يقسم مساحة الجهاز إلى وحدات metaslab وادارتها (يعطي volume بإمكانه احتواء المئات من metaslabs).
   j) كل meatslab تكون معرفة مع space map (يستخدم في الحسابات الخوارزمية).
   k) يسجل إلى log file بدلا من نظام الملف (log of all block activity, in time order, in counting format).
   l) نشاط Metaslab يخزن مساحة الخريطة إلى الذاكرة في هيكلة الشجرة الموزونة ومفهرسة عن طريق التعويض:
      - إعادة log إلى تلك الهيكلة.
      - ربط free blocks المتجاورة إلى مدخل واحد.

39: **هناك انواع تقنية متنوعة لتحسين أداء وكفاءة وحدة التخزين الثانوي (*Efficiency and Performance*):**

- الكفاءة (***Efficiency***) تعتمد على:

1. تخصيصي القرص وخوارزميات الدليل.
2. أنواع البيانات تحتفظ في مدخل دليل الملف.
3. قبل التخصيص او تخصيص ما هو مطلوب من هياكل ال metadata.
4. الحجم الثابت او الاحجام المتفاوتة من هياكل البيانات.

- الاداء (***Performance***):

1. الاحتفاظ بالبيانات وال metadata قريبة من بعضها البعض.
2. Buffer Cache: يفصل قسم من الذاكرة الرئيسية لاستخدام blocks بشكل مستمر.
3. التزامن (Synchronous): يكتب في بعض المرات طلبات بواسطة apps أو الاحتياجات بواسطة OS:
   a) No buffering/caching يكتب وجوب استهداف القرص قبل الاشعار بالاستلام.
   b) غير متزامن (Asynchronous) يكتب الاكثر شيوعا، buffer-able ، faster.
4. Free-behind and read-ahead: هي تقنيات لتحسين الوصول المتسلسل.
5. القراءات المستمرة أبطأ من الكتابات.

40: **Buffer Cache:** يفصل قسم من الذاكرة الرئيسية لاستخدام blocks بشكل مستمر.

41: **الفرق بين الاداء المتزامن والاداء الغير متزامن في وحدة التخزين:**
- **المتزامن (Synchronous):** يكتب في بعض المرات طلبات بواسطة apps أو الاحتياجات بواسطة OS.
- **الغير متزامن (Asynchronous):** يكتب الاكثر شيوعا، buffer-able ، faster.

42: **Page Cache:** caches pages بدلا disk blocks باستخدام تقنيات الذاكرة الافتراضية والعناوين.
- Memory-mapped I/O يستخدم page cache.
- Routing I/O من خلال نظام الملف يستخدم the buffer (disk) cache.

43: **Unified buffer cache:** يستخدم نفس page cache إلى cache صفحات ال memory-mapped و I/O نظام الملف معا لتجنب double caching.

44: **الاصلاح او الاستعادة (Recovery):** ملفات ودلائل تكون محفوظة معا في الذاكرة الرئيسية بما يجب اتخاذه للتأكد من ان فشل النظام لا يؤدي فقدان البيانات أو في عدم تناسق (inconsistency) البيانات.

45: **الفحص المتناسق (Consistency checking):** يقوم بمقارنة البيانات في هيكلة الدليل مع بيانات ال blocks على القرص، ويحاول اصلاح الغير متناسق.

46: **الاصلاح (recovery)** يستخدم برمج النظام لدعم (buck up) البيانات من القرص إلى جهاز تخزين آخر مثل الشريط الممغنظ والاقراص الممغنطة الاخرى والبصريات.

47: استعادة (recover) الملفات المفقودة او القرص عن طريق استعادة (restoring) البيانات من ال backup.

48: **السجل المنظم (Log structured):** السجل المنظم او ما يسمى بـ journaling يقوم بتسجيل كل تحديثات metadata إلى نظام الملف كـ معاملة (transaction).

49: **من خصائص تعاملات السجل المنظم (transactions of log structured):**

1. كل التعاملات تكون مكتوبة في السجل (log):
   - Transaction تكون ملتزمة بمجرد كتابتها في السجل (بالتتابع).
   - بعض المرات تفصل الجهاز او تجزئ القرص.
   - مع ذلك، في بعض الاحيان تكون ملف النظام لم يحدث بعد.
2. المعاملات في السجل (log) تكون مكتوبة بشكل غير متزامن في هياكل نظم الملفات.
   - عندما يتم تعديل هياكل نظام الملف، المعاملة تحذف من السجل (log).
3. عند انهيار ملفات النظام، جميع التعاملات المتبقية في السجل يجب أن تكون في طور التنفيذ.
4. الاستعادة السريعة من الانهيار يزيل فرص ال inconsistency of metadata.

50: **المعاملات في السجل (log)** تكون مكتوبة بشكل غير متزامن في هياكل نظم الملفات. وتحذف بعد تعديل هياكل نظام الملف.

51: **أنظمة ملفات الشبكة (NFS: Network file systems):** أنظمة ملفات النظام تكون عادة مدمجة مع هيكلة الدليل الشامل وواجهة نظام العميل. وهو لتنفيذ وتخصيص (implementation and specification) نظام السوفتوير للوصول إلى الملفات عن بعد عبر LAN or WAN.

52: **تنفيذ أنظمة ملفات الشبكة** هو جزء من Solaris وأنظمة التشغيل الشمسي التي تعمل على المحطات الشمسية باستخدام بروتوكول Datagram الغير فعالة.

53: **الدليل عن بعد (Remote directory):** يتم تركيبه على دليل نظام الملفات المحلية.

54: **نظام ملفات الشبكة (NFS)** مصممة للتشغيل في البيئة الغير متجانسة (heterogeneous environment) لمختلف الالات، ولنظم التشغيل، وايضا أبنية الشبكة (network architectures).

55: **تخصيص (specification) نظام ملفات الشبكة** يقوم بالتمييز (distinguish) بين الخدمات المتوفرة وذلك عن طريق مقدار الآلية وخدمات الوصول فعليا إلى الملفات عن بعد.

56: **وظائف بروتوكول انظمة ملفات الشبكة (NFS Mount Protocol):**

1. يؤسس علاقة منطقية اولية بين الخادم (Server) والعميل (Client).
2. عملية الـ mount تشمل اسم الدليل عن بعد لتكون مركبه (to be mounted) واسم آلة الخادم مخزنة فيها.
3. اتباع طلب الـ a mount الذي يتوافق مع قائمة صادراتها، الخادم يعيد الـ file handle.
4. عملية الـ mount يغير فقط عرض المستخدم ولا يؤثر في جهة الخادم.

57: **عملية الـ mount** تشمل اسم الدليل عن بعد لتكون مركبه (to be mounted) واسم آلة الخادم مخزنة فيها.

58: **قائمة التصدير (Export List)** يحدد انظمة الملفات المحلية التي تجعل الخادم يصدر للـ mounting جنبا إلى جنب مع اسماء الالات التي تسمح بتركيبها (to mount them).

59: **File Handle:** هو معرف (identifier) نظام الملفات وعدد الـ inode لتعريف الدليل المركب في نظام الملفات المصدرة.

60: **عملية التركيب (the mount operation)** يغير فقط عرض المستخدم ولا يؤثر في جهة الخادم.

61: **بروتوكول (NFS)** يقوم بتوفير مجموعة من اجراءات النداءات عن بعد لعمليات الملفات عن بعد.

62: **تدعم اجراءات عمليات الملفات عن بعد بواسطة العمليات التالية:**

1. البحث عن ملف في الدليل.
2. قراءة مجموعة من مدخلات الدليل.
3. يعالج (manipulating) الروابط والدلائل.
4. الوصول إلى خصائص الملفات.
5. قراءة وكتابة الملفات.

63: **خوادم انظمة ملفات الشبكة (NFS Servers)** هي غير واضحة (stateless) وكل طلب يحتاج توفير مجموعة كاملة من الحجج (arguments).

64: **بروتوكول NFS** لا يوفر آليات التحكم المتوافقه (Concurrency-control).

65: **الطبقات الرئيسية لبنية أنظمة ملفات الشبكة (Layers of NFS Architecture):**

1. **واجهة نظام ملف الـ Unix** مبني على نداءات الفتح والقراءة والكتابة والاغلاق وأيضا واصفات الملف (file descriptors).
2. **طبقة نظام الملف الافتراضي (VFS Layer)** يقوم بتمييز الملفات المحلية من الملفات عن بعد، والملفات المحلية تتميز وفقا لأنواع نظام الملفات الخاصة بها.
3. **طبقة خدمة NFS:** يقع في اخر طبقات البنية (layer of the architecture) والذي يقوم بتنفيذ بروتوكول الـ NFS.

66: **ترجمة اسم المسار لأنظمة ملفات الشبكة (NFS Path-Name Translation)** يقوم بعمل كسر المسار إلى أسماء مكونات ويقوم بفصل نداء البحث NFS لكل زوج من اسم المحتوى و vnode الدليل.

67: **لجعل البحث (lookup) أسرع**، مخبأ (cache) بحث اسم الدليل الموجود على جهة العميل يقوم بالتمسك بـ vnodes لأسماء الدليل عن بعد.

68: **File-blocks cache** عندما يكون الملف مفتوح، والنواة () يقوم بفحص الخادم عن بعد سواءا لجلب أو إعادة تأكيد سمات الـ cached.

69: **Cached file blocks** يستخدم فقط لو كان سمات الـcached متطابقة حتى الآن.

END of chapter 11