جَـامعة
المَـنارة
MANARA UNIVERSITY

Advanced Operating System

Lecture Notes

Dr. Professor, J.M. Khalifeh

قسم المعلوماتية

الرابعة

Unit-4

Mass storage systems

ملخص

إن نظام التخزين الشامل الرئيسي في أجهزة الكمبيوتر الحديثة هو التخزين الثانوي، والذي يتم توفيره عادةً بواسطة محركات الأقراص الصلبة hard disk drives (HDD) وأجهزة الذاكرة غير المتطايرة nonvolatile memory (NVM) . كما تحتوي بعض الأنظمة على تخزين ثانوي أبطأ وأكبر حجمًا، ويتكون عمومًا من شريط مغناطيسي أو أقراص ضوئية أو حتى تخزين سحابي. ولأن أجهزة التخزين الأكثر شيوعًا وأهمية في أنظمة الكمبيوتر الحديثة هي محركات الأقراص الصلبة وأجهزة الذاكرة غير المتطايرة، فإن الجزء الأكبر من هذه الوحدة مخصص لمناقشة هذين النوعين من التخزين. سنعتبر هنا أن الطالب قد تعرف على البنية الفيزيائية لهذه الأنظمة وسنركز هنا على خوارزميات الجدولة، التي تقوم بجدولة ترتيب عمليات الإدخال/الإخراج لتحقيق أقصى قدر من الأداء. بعد ذلك، نناقش بنية أنظمة RAID. هناك العديد من أنواع التخزين الشامل، ونستخدم المصطلح العام للتخزين غير المتطاير ((NVS أو نتحدث عن "أقراص" التخزين عندما تشمل المناقشة جميع الأنواع.

أهداف الوحدة

- شرح خصائص أداء أجهزة التخزين الشامل.
- تقييم خوارزميات جدولة الإدخال/الإخراج.
- مناقشة خدمات نظام التشغيل المقدمة لنظام التخزين الشامل، بما في ذلك **RAID**.

جدولة الأقراص Disk Scheduling

تتمتع الأقراص المغناطيسية التقليدية بالبنية الأساسية التالية:

قرص واحد أو أكثر على شكل أقراص مغطاة بوسائط مغناطيسية. تصنع أقراص القرص الصلب من معدن صلب، في حين تصنع الأقراص "المرنة" من بلاستيك أكثر مرونة.

يحتوي كل قرص على سطحين للعمل. في بعض الأحيان، لا تستخدم محركات الأقراص الصلبة القديمة السطح العلوي أو السفلي لمجموعة من الأقراص، حيث تكون هذه الأسطح أكثر عرضة للتلف المحتمل.

ينقسم كل سطح عمل إلى عدد من الحلقات المتحدة المركز تسمى المسارات. وتسمى مجموعة المسارات التي تكون على نفس المسافة من حافة القرص (أي جميع المسارات الموجودة مباشرة فوق بعضها البعض في الرسم التخطيطي التالي) أسطوانة.

ينقسم كل مسار إلى قطاعات، تحتوي تقليديًا على 512 بايت من البيانات لكل منها، على الرغم من أن بعض الأقراص الحديثة تستخدم أحيانًا أحجام قطاعات أكبر. (تتضمن القطاعات أيضًا رأسًا ومقطورة، بما في ذلك معلومات المجموع الاختباري من بين أشياء أخرى. تقلل أحجام القطاعات الأكبر من نسبة القرص المستهلكة بواسطة الرؤوس والمقطورات، ولكنها تزيد من التفتت الداخلي وكمية القرص التي يجب وضع علامة عليها بأنها تالفة في حالة وجود أخطاء.)

يتم قراءة البيانات الموجودة على القرص الصلب بواسطة رؤوس القراءة والكتابة. يستخدم التكوين القياسي (الموضح أدناه) رأسًا واحدًا لكل سطح، كل منها على ذراع منفصل، ويتم التحكم فيه بواسطة مجموعة ذراع مشتركة تنقل جميع الرؤوس في وقت واحد من أسطوانة إلى أخرى. (قد تعمل التكوينات الأخرى، بما في ذلك رؤوس القراءة والكتابة المستقلة، على تسريع الوصول إلى القرص، ولكنها تنطوي على صعوبات فنية خطيرة.)

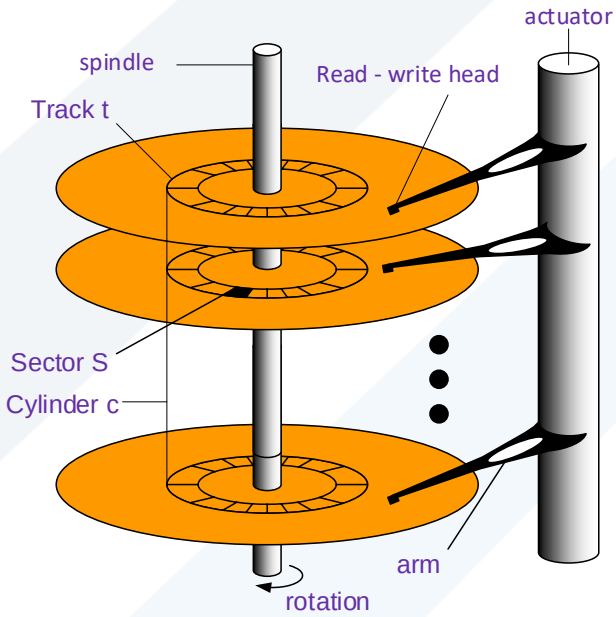
تساوي سعة تخزين محرك الأقراص التقليدي عدد الرؤوس (أي عدد الأسطح العاملة)، مضروبًا في عدد المسارات لكل سطح، مضروبًا في عدد القطاعات لكل مسار، مضروبًا في عدد البايتات لكل قطاع. يتم تحديد كتلة مادية معينة من البيانات من خلال توفير رقم الرأس والقطاع والأسطوانة التي توجد بها.

تعتبر خوارزميات جدولة القرص حاسمة في إدارة كيفية قراءة البيانات من القرص الصلب للكمبيوتر وكتابتها عليه. تساعد هذه الخوارزميات في تحديد الترتيب الذي تتم به معالجة طلبات القراءة والكتابة على القرص، مما يؤثر بشكل كبير على سرعة وكفاءة الوصول إلى البيانات. تشمل طرق جدولة القرص الشائعة أسبقية الحضور، أسبقية الخدمة ((FCFS، وأقصر وقت بحث أولاً ((SSTF، وSCAN، وC-SCAN، وLOOK، و .C-LOOKمن خلال فهم هذه الخوارزميات وتنفيذها، يمكننا تحسين أداء النظام وضمان استرجاع البيانات بشكل أسرع.

جدولة القرص هي تقنية تستخدمها أنظمة التشغيل لإدارة الترتيب الذي تتم به معالجة طلبات الإدخال/الإخراج على القرص. تُعرف جدولة القرص أيضًا باسم جدولة الإدخال/الإخراج.

الأهداف الرئيسية لجدولة القرص هي تحسين أداء عمليات القرص، وتقليل الوقت المستغرق للوصول إلى البيانات وتحسين كفاءة النظام بشكل عام.

وهنا سوف نستكشف الأنواع المختلفة من خوارزميات جدولة القرص ووظائفها. من خلال فهم هذه الخوارزميات وتنفيذها، يمكننا تحسين أداء النظام وضمان استرجاع البيانات بشكل أسرع.



أهمية جدولة القرص في نظام التشغيل

قد تصل طلبات إدخال/إخراج متعددة من خلال عمليات مختلفة، ولا يمكن تقديم سوى طلب إدخال/إخراج واحد في كل مرة بواسطة وحدة تحكم القرص. وبالتالي، تحتاج طلبات الإدخال/الإخراج الأخرى إلى الانتظار في قائمة الانتظار وتحتاج إلى جدولة.

قد يكون هناك طلبان أو أكثر بعيدين عن بعضهما البعض، مما قد يؤدي إلى زيادة حركة ذراع القرص.

تعتبر محركات الأقراص الصلبة واحدة من أبطأ أجزاء نظام الكمبيوتر، وبالتالي يجب الوصول إليها بطريقة فعّالة.

المصطلحات الرئيسية المرتبطة بجدولة القرص

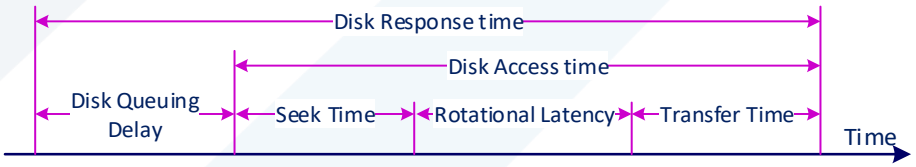
- زمن جدولة القرص **Disk Queuing Time**: يشير طول قائمة انتظار القرص إلى عدد العمليات أو الطلبات المعلقة التي تنتظر معالجتها بواسطة وحدة تخزين معينة. وهو مؤشر على حجم العمل على القرص ويمكن أن يساعد في تحديد ما إذا كانت الوحدة قادرة على التعامل مع معدل الطلب بكفاءة. ويكون زمن جدولة القرص هو الزمن الذي تقضيه العمليات في طابور انتظار القرص قبل الاستجابة لها.

- زمن البحث **Seek Time**: وقت البحث هو الوقت المستغرق لتحديد موقع ذراع القرص على مسار محدد حيث سيتم قراءة البيانات أو كتابتها. لذا فإن خوارزمية جدولة القرص التي تعطي الحد الأدنى من متوسط وقت البحث هي الأفضل.
- زمن الانتقال الدوراني **Rotational Latency**: زمن الانتقال الدوراني هو الوقت المستغرق من قبل القطاع المطلوب من القرص للدوران إلى موضع بحيث يمكنه الوصول إلى رؤوس القراءة/الكتابة. لذا فإن خوارزمية جدولة القرص التي تعطي الحد الأدنى من زمن الانتقال الدوراني هي الأفضل.
- زمن النقل **Transfer Time**: وقت النقل هو الوقت اللازم لنقل البيانات. ويعتمد على سرعة دوران القرص وعدد البايتات المراد نقلها.
- زمن الوصول إلى القرص **Disk Access Time**:

زمن الوصول إلى القرص = زمن البحث + زمن الانتقال الدوراني + زمن النقل

إجمالي زمن البحث = إجمالي حركة الرأس * زمن البحث

زمن استجابة القرص Disk Response Time: زمن الاستجابة هو متوسط الوقت الذي يقضيه الطلب في انتظار تنفيذ عملية الإدخال/الإخراج. ومتوسط زمن الاستجابة هو زمن استجابة جميع الطلبات. وزمن استجابة التباين هو مقياس لكيفية خدمة الطلبات الفردية فيما يتعلق بمتوسط زمن الاستجابة. لذا فإن خوارزمية جدولة القرص التي توفر الحد الأدنى من زمن استجابة التباين هي الأفضل.



هدف خوارزميات جدولة القرص

- تقليل وقت البحث
- تعظيم الإنتاجية
- تقليل زمن الوصول
- العدالة
- الكفاءة في استخدام الموارد

أول من يأتي أول من يخدم **FCFS**

يعتبر FCFS أبسط خوارزميات جدولة القرص على الإطلاق. في FCFS، تتم معالجة الطلبات بالترتيب الذي تصل به إلى قائمة انتظار القرص. دعنا نفهم هذا بمساعدة مثال.

مثال:

افترض أن ترتيب الطلب هو – (82,170,43,140,24,16,190)

والموضع الحالي لرأس القراءة/الكتابة هو: 50

لذا، فإن المسافة الإجمالية التي يغطيها ذراع القرص =

642= (16−190)+(16−24)+(24−140)+(43−140)+(43−170)+(82−170)+(50−82)

مزايا نظام FCFS

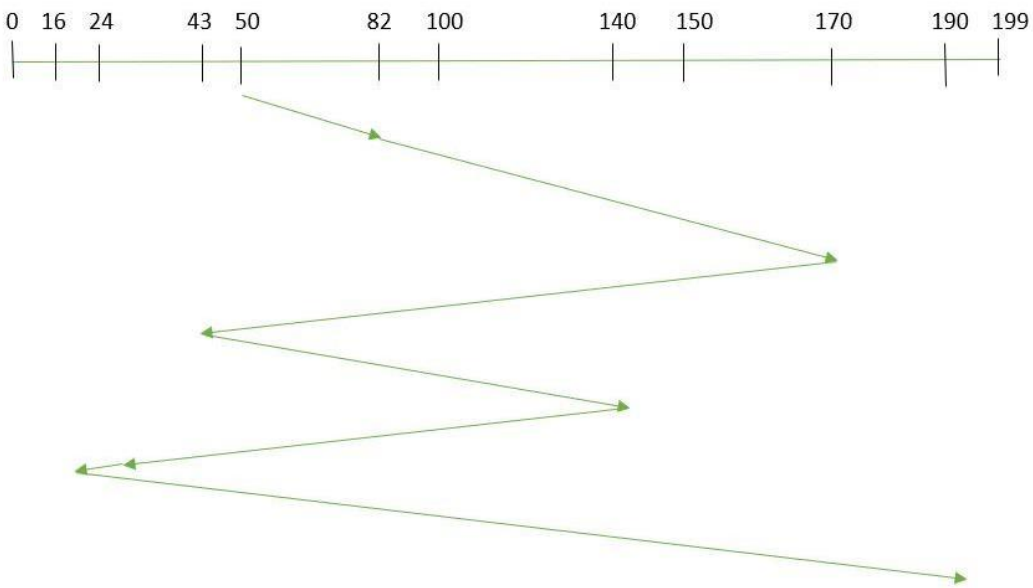
فيما يلي بعض مزايا نظام First Come First Serve.

- كل طلب يحصل على فرصة عادلة
- لا يوجد تأجيل غير محدد

عيوب نظام FCFS First Come First Serve

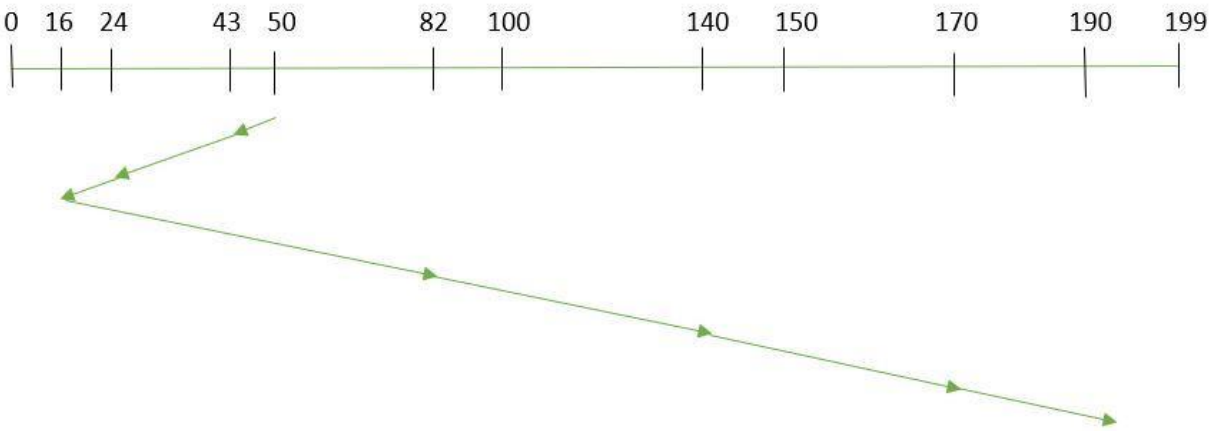
فيما يلي بعض عيوب نظام FCFS

- لا تحاول تحسين وقت البحث
- قد لا تقدم أفضل خدمة ممكنة



2. أقصر وقت بحث أولا Shortest Seek Time First SSTF

في SSTF يتم تنفيذ الطلبات التي لها أقصر وقت بحث أولاً. لذا، يتم حساب وقت البحث لكل طلب مسبقًا في قائمة الانتظار ثم يتم جدولتها وفقًا لوقت البحث المحسوب لها. ونتيجة لذلك، سيتم تنفيذ الطلب القريب من ذراع القرص أولاً. يعد SSTF بالتأكيد تحسينًا على FCFS لأنه يقلل من متوسط وقت الاستجابة ويزيد من إنتاجية النظام. دعنا نفهم هذا بمساعدة مثال.



افترض أن ترتيب الطلب هو- (82,170,43,140,24,16,190)

والموضع الحالي لرأس القراءة/الكتابة هو: 50

لذا،

إجمالي الحركة العلوية (إجمالي المسافة التي يغطيها ذراع القرص) =

208= (170−190)+(140−170)+(82−140)+(16−82)+(16−24)+(24−43)+(43−50)

مزايا أقصر وقت بحث أولاً

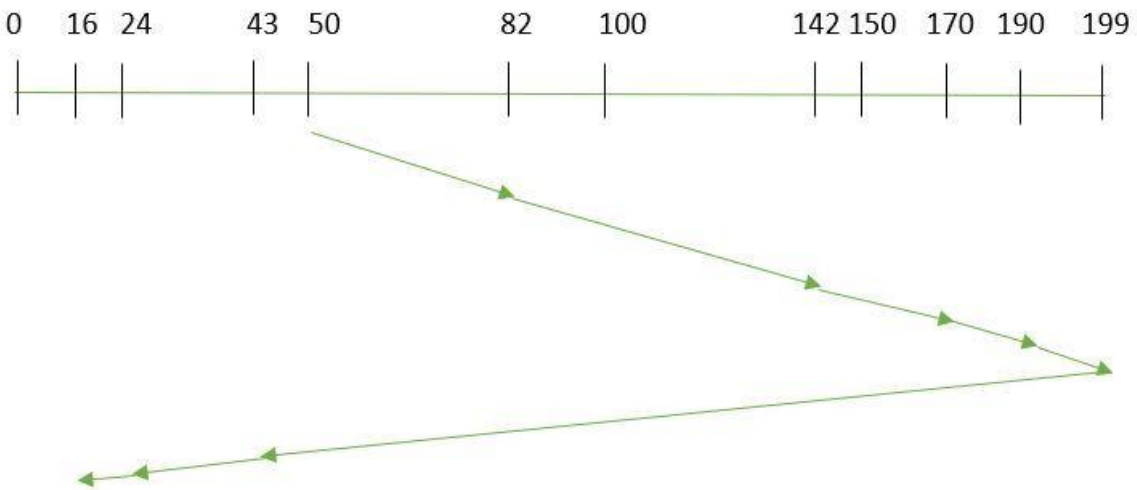
- يقل متوسط وقت الاستجابة
- يزداد الإنتاج

عيوب أقصر وقت بحث أولاً

- التكاليف الإضافية لحساب وقت البحث مقدمًا
- يمكن أن يتسبب في نقص الطلب إذا كان وقت البحث فيه أعلى مقارنة بالطلبات الواردة
- التباين العالي لوقت الاستجابة حيث يفضل **SSTF** بعض الطلبات فقط

3. SCAN

في خوارزمية SCAN، يتحرك ذراع القرص في اتجاه معين ويخدم الطلبات الواردة في مساره وبعد الوصول إلى نهاية القرص، يعكس اتجاهه ويخدم مرة أخرى الطلب الذي يصل في مساره. لذا، تعمل هذه الخوارزمية كمصعد وبالتالي تُعرف أيضًا باسم خوارزمية المصعد. ونتيجة لذلك، تتم خدمة الطلبات في النطاق المتوسط بشكل أكبر وسيتعين على الطلبات التي تصل خلف ذراع القرص الانتظار.



افترض أن الطلبات التي يجب معالجتها هي 190-،16،24،140،43،170،82. وأن ذراع القراءة/الكتابة عند 50، ومن المعلوم أيضًا أن ذراع القرص يجب أن تتحرك "باتجاه القيمة الأكبر".

لذلك، يتم حساب إجمالي الحركة العلوية (المسافة الإجمالية التي يغطيها ذراع القرص) على النحو التالي

$$= (50-199) + (16-199) = 332$$

مزايا خوارزمية SCAN

- إنتاجية عالية
- تباين منخفض في زمن الاستجابة
- متوسط زمن الاستجابة

عيوب خوارزمية SCAN

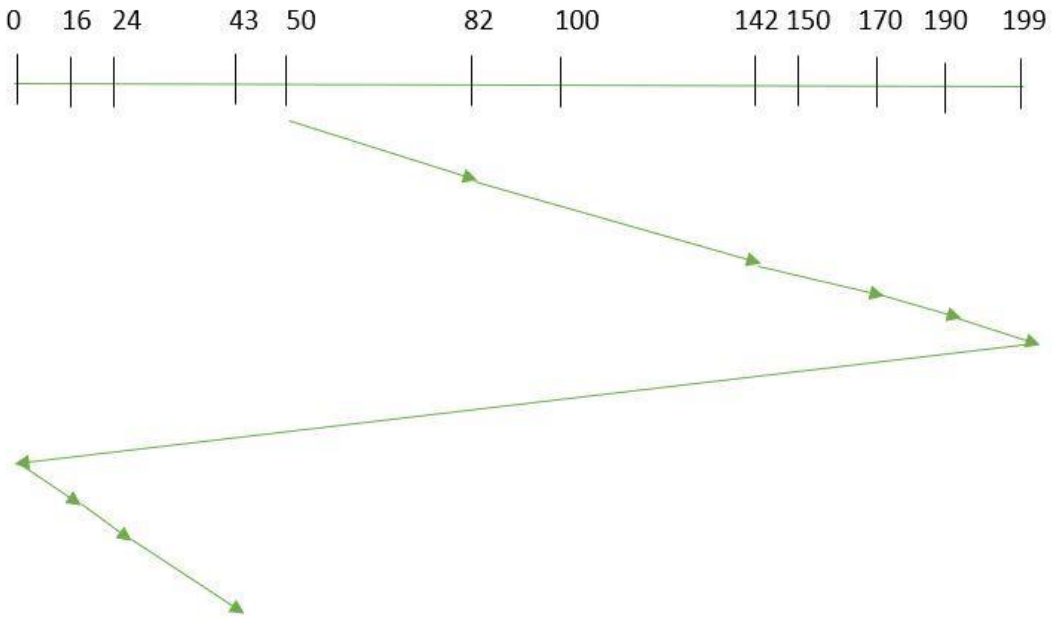
- وقت انتظار طويل لطلبات المواقع التي زارها ذراع القرص للتو

4. C-SCAN

في خوارزمية SCAN، يقوم ذراع القرص مرة أخرى بمسح المسار الذي تم مسحه، بعد عكس اتجاهه. لذا، قد يكون من الممكن أن يكون هناك عدد كبير جدًا من الطلبات تنتظر في الطرف الآخر أو قد يكون هناك عدد قليل أو صفر من الطلبات المعلقة في المنطقة الممسوحة ضوئيًا.

يتم تجنب هذه المواقف في خوارزمية CSCAN حيث يذهب ذراع القرص بدلاً من عكس اتجاهه إلى الطرف الآخر من القرص ويبدأ في خدمة الطلبات من هناك. لذا، يتحرك ذراع القرص بطريقة دائرية وهذه الخوارزمية مماثلة أيضًا لخوارزمية SCAN ومن ثم تُعرف باسم) C-SCANالمسح الدائري).

مثال:



افترض أن الطلبات التي يجب معالجتها هي 82،170،43،140،24،16،190–. وأن ذراع القراءة/الكتابة عند 50، ومن المعلوم أيضًا أن ذراع القرص يجب أن تتحرك "باتجاه القيمة الأكبر".

لذا، يتم حساب إجمالي الحركة العلوية (المسافة الإجمالية التي يغطيها ذراع القرص) على النحو التالي:

391 = (43-0) + (199-0) + (199-50)=

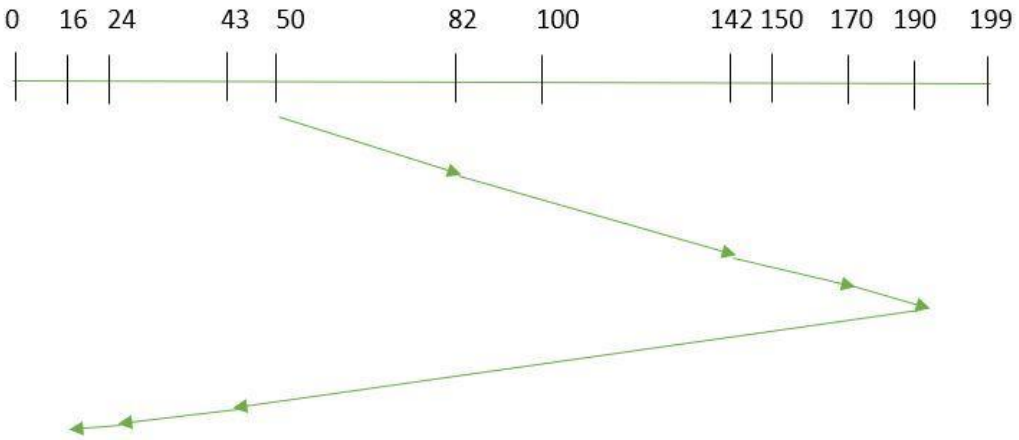
مزايا خوارزمية C-SCAN

توفر وقت انتظار أكثر انتظامًا مقارنةً بخوارزمية SCAN.

5. LOOK

خوارزمية LOOK تشبه خوارزمية جدولة قرص SCAN باستثناء الاختلاف الذي يتمثل في أن ذراع القرص على الرغم من انتقاله إلى نهاية القرص، فإنه يذهب فقط إلى آخر طلب يتم خدمته أمام الرأس ثم يعكس اتجاهه من هناك فقط. وبالتالي، فإنه يمنع التأخير الإضافي الذي يحدث بسبب الانتقال غير الضروري إلى نهاية القرص.

مثال:



لنفترض أن الطلبات المطلوب معالجتها هي 82،170،43،140،24،16،190-. وأن ذراع القراءة/الكتابة عند 50، ومن المعلوم أيضًا أن ذراع القرص يجب أن تتحرك "باتجاه القيمة الأكبر".

لذا، يتم حساب إجمالي الحركة العلوية (المسافة الإجمالية التي يغطيها ذراع القرص) على النحو التالي:

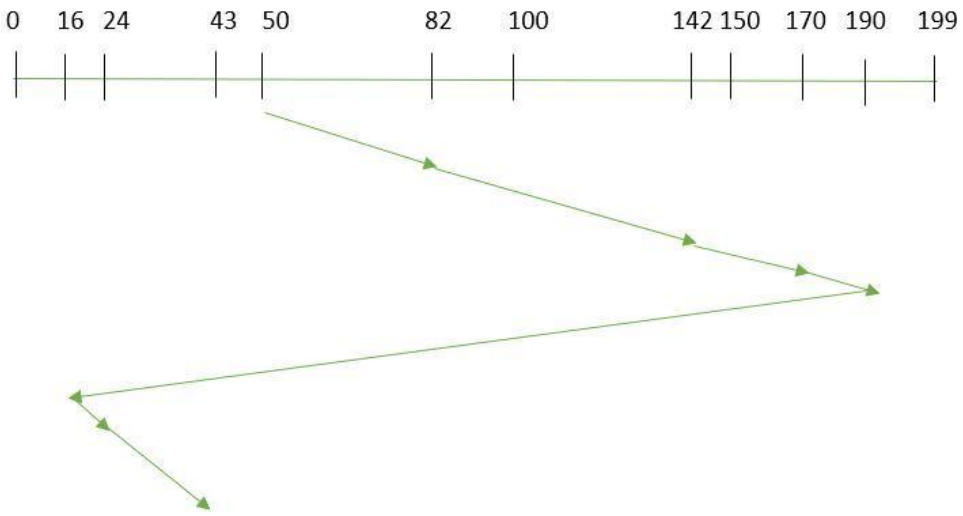
314 = (16-190) + (50-190) =

6. C-LOOK

نظرًا لأن LOOK مشابه لخوارزمية SCAN، فإن C-LOOK مشابه بطريقة مماثلة لخوارزمية جدولة القرص CSCAN. في CLOOK، على الرغم من ذهاب ذراع القرص إلى النهاية، فإنه يذهب فقط إلى آخر طلب يتم خدمته أمام الرأس ثم ينتقل من هناك إلى آخر طلب للطرف الآخر. وبالتالي، فإنه يمنع أيضًا التأخير الإضافي الذي حدث بسبب الانتقال غير الضروري إلى نهاية القرص.

مثال:

افترض أن الطلبات التي سيتم معالجتها هي82,170,43,140,24,16,190-. وأن ذراع القراءة/الكتابة عند 50، ومن المعلوم أيضًا أن ذراع القرص يجب أن تتحرك "نحو القيمة الأكبر"



لذا، يتم حساب إجمالي الحركة العلوية (المسافة الإجمالية التي يغطيها ذراع القرص) على النحو التالي

= (50−190) + (16−190) + (16−43) = 341

RAID مصفوفات مكررة من الأقراص المستقلة Redundant Arrays of Independent Disks

RAID هي تقنية تستخدم مجموعة من الأقراص المتعددة لتخزين البيانات بدلاً من استخدام قرص واحد، وذلك لتحسين الأداء أو تكرار البيانات أو لحماية البيانات في حالة فشل محرك الأقراص.

يشبه RAID الاحتفاظ بنسخ احتياطية من ملفاتك المهمة مخزنة في أماكن مختلفة على العديد من محركات الأقراص الصلبة أو محركات الأقراص ذات الحالة الصلبة SSDs إذا توقف أحد المحركات عن العمل، فستظل بياناتك آمنة لأن لديك نسخًا أخرى مخزنة على محركات الأقراص الأخرى. إنه مثل وجود شبكة أمان لحماية ملفاتك من الضياع إذا تعطل أحد محركات الأقراص لديك.

يعتبر RAID مصفوفات مكررة من الأقراص المستقلة في نظام إدارة قواعد البيانات DBMS تقنية تجمع بين محركات أقراص مادية متعددة في وحدة منطقية واحدة لتخزين البيانات. الغرض الرئيسي من RAID هو تحسين موثوقية البيانات وتوافرها وأدائها. هناك مستويات مختلفة من RAID، كل منها يوفر توازنًا بين هذه الفوائد.

نقاط التقييم الرئيسية لنظام RAID

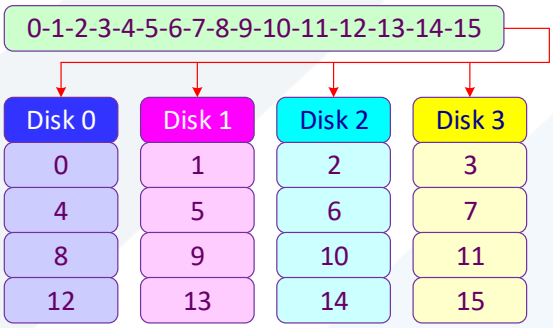
عند تقييم نظام RAID، يجب مراعاة الجوانب المهمة التالية:

1. الموثوقية Reliability: تشير إلى قدرة النظام على تحمل أخطاء القرص ومنع فقدان البيانات.
2. التوافرية Availability: جزء الوقت الذي يكون فيه نظام RAID قيد التشغيل ومتاحًا للاستخدام.
3. الأداء Performance: يقيس مدى كفاءة نظام RAID في التعامل مع مهام معالجة البيانات. ويشمل ذلك:
   - زمن الاستجابة Response Time : مدى سرعة استجابة النظام لطلبات البيانات.
   - الإنتاجية Throughput: المعدل الذي يعالج به النظام البيانات (على سبيل المثال، ميجابايت/ثانية أو IOPS).

4. السعة Capacity: مقدار التخزين القابل للاستخدام المتاح للمستخدم بعد مراعاة آليات التكرار.

1. RAID-0

يعمل RAID–0 على تحسين أداء النظام من خلال تقسيم البيانات إلى "كتل" أصغر ونشرها عبر أقراص متعددة. وتعزز هذه الطريقة سرعة الوصول إلى البيانات من خلال تمكين عمليات القراءة/الكتابة المتوازية ولكنها لا توفر التكرار أو التسامح مع الأخطاء.



- تشكل مجموعة الكتل الموزعة عبر الأقراص "شريطًا". على سبيل المثال، "0، 1، 2، 3" عبارة عن شريط واحد.
- بدلاً من وضع كتلة واحدة فقط في قرص في كل مرة، يمكننا العمل مع كتلتين (أو أكثر).

التقييم

الموثوقية:

لا يوجد تكرار للبيانات. وبالتالي، لا يمكن استعادة الكتلة المفقودة.

السعة: *N*B*

يتم استخدام المساحة بالكامل لتخزين البيانات. نظرًا لعدم وجود تكرار، يتم الاستفادة الكاملة من N قرصًا يحتوي كل منها على B كتلة.

المزايا

- سهل التنفيذ.
- يستخدم سعة التخزين بطريقة أفضل.

العيوب

- يمكن أن يؤدي فقدان محرك أقراص واحد إلى فشل النظام بالكامل.
- إنه ليس خيارًا جيدًا لنظام بالغ الأهمية.

2. RAID-1 (النسخ المتطابق)

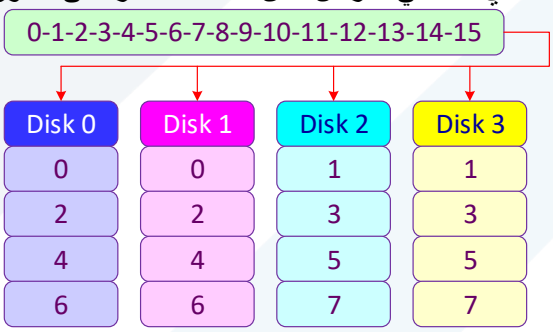
يعزز RAID-1 الموثوقية من خلال إنشاء نسخة متطابقة (نسخة متطابقة) لكل كتلة بيانات على أقراص منفصلة. وهذا يضمن أنه حتى في حالة فشل أحد الأقراص، تظل البيانات قابلة للوصول من نسخته المكررة. في حين أن هذا التكوين موثوق للغاية، إلا أنه يتطلب تكلفة تخزين كبيرة.

على سبيل المثال:

- يتم تخزين الكتلة 0 على القرص 0 ونسختها المكررة على القرص 1.
- يتم تخزين الكتلة 1 على القرص 2 ونسختها المكررة على القرص 3.

يشير الشكل إلى وجود نسختين من كل كتلة، موزعة على أقراص مختلفة.

لم يكن RAID 0 قادرًا على تحمل أي فشل في القرص. لكن RAID 1 قادر على تجاوز الفشل في أحد الأقراص أو أكثر.



التقييم

الموثوقية: 1 إلى N/2

يمكن التعامل مع فشل قرص واحد بالتأكيد لأن كتل ذلك القرص سيكون لها نسخ مكررة على قرص آخر. إذا كنا محظوظين بما فيه الكفاية وفشل القرصان 0 و2، فيمكن التعامل مع هذا مرة أخرى لأن كتل هذه الأقراص بها نسخ مكررة على القرصين 1 و3. لذا، في أفضل الأحوال، يمكن التعامل مع فشل قرصين .N/2

السعة: N*B/2

يتم استخدام نصف المساحة فقط لتخزين البيانات. والنصف الآخر عبارة عن نسخة طبق الأصل من البيانات المخزنة بالفعل.

المزايا

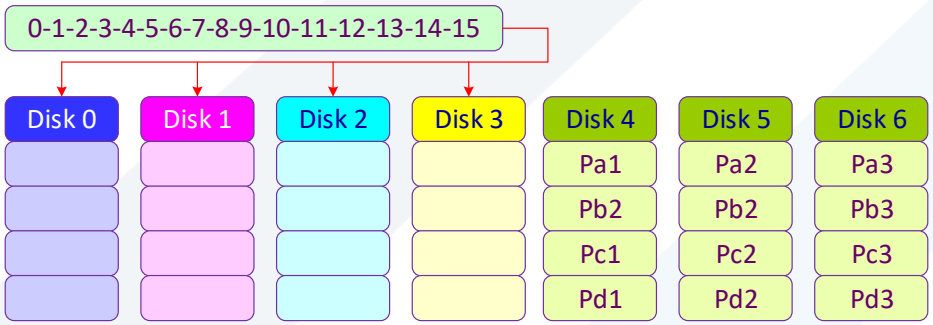
- يغطي التكرار الكامل.
- يمكنه زيادة أمان البيانات وسرعتها.

العيوب

- مكلف للغاية.
- سعة التخزين أقل.

3. RAID-2 (تجريد مستوى البت مع التكافؤ المخصص)

RAID-2 هو مستوى RAID متخصص يستخدم التجريد على مستوى البت مع تصحيح الأخطاء باستخدام شفرة هامينج. في هذا التكوين، يتم توزيع البيانات على مستوى البت عبر محركات متعددة، ويتم استخدام محرك تكافؤ مخصص للكشف عن الأخطاء وتصحيحها. في حين أنه يوفر تسامحًا قويًا مع الأخطاء، إلا أن تعقيده وتكلفته تجعله نادرًا ما يستخدم في الممارسة العملية.



التقييم

المزايا

- في حالة تصحيح الأخطاء، يستخدم شفرة هامينج.
- يستخدم محرك أقراص مخصص لتخزين التكافؤ.

العيوب

- له بنية معقدة وتكلفة عالية بسبب المحرك الإضافي.
- يتطلب محرك أقراص إضافيًا للكشف عن الأخطاء.

4. RAID-3 (التجزئة على مستوى البايت مع التكافؤ المخصص)

يعزز RAID-3 تحمل الأخطاء من خلال استخدام التجزئة على مستوى البايت عبر محركات أقراص متعددة وتخزين معلومات التكافؤ على محرك أقراص تكافؤ مخصص. يسمح محرك الأقراص التكافؤ المخصص بإعادة بناء البيانات المفقودة في حالة فشل محرك أقراص واحد. هذا التكوين مناسب لأحمال العمل التي تتطلب إنتاجية عالية للبيانات المتسلسلة ولكنه أقل كفاءة لعمليات الإدخال/الإخراج العشوائية.

في مثالنا يحتوي القرص 3 على بتات التكافؤ للقرص 0 والقرص 1 والقرص 2. إذا حدث فقد للبيانات، يمكننا إنشائه باستخدام القرص 3.

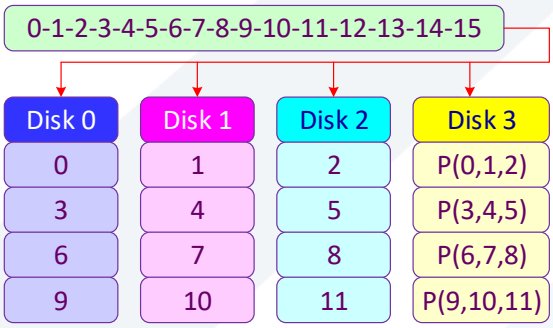
التقييم

الموثوقية:

يمكن لـ RAID-3 تحمل فشل قرص واحد. يمكن إعادة بناء البيانات المفقودة باستخدام محرك التكافؤ ومحركات البيانات المتبقية.

السعة:

السعة القابلة للاستخدام = $(N-1) \times B$ حيث N هو العدد الإجمالي للمحركات، وB هو عدد الكتل لكل محرك. يتم حجز سعة محرك واحد لتخزين معلومات التكافؤ.



المزايا

- يمكن نقل البيانات بكميات كبيرة.
- يمكن الوصول إلى البيانات بالتوازي.

العيوب

- يتطلب محركًا إضافيًا للتكافؤ.
- في حالة الملفات ذات الحجم الصغير، يعمل ببطء.

5. RAID-4 (التجزئة على مستوى الكتلة باستخدام التكافؤ المخصص)

يقدم RAID-4 التجزئة على مستوى الكتلة عبر أقراص متعددة، جنبًا إلى جنب مع قرص تكافؤ مخصص لتوفير التسامح مع الأخطاء. تتم كتابة البيانات في كتل، ويخزن قرص منفصل معلومات التكافؤ المحسوبة باستخدام دالة XOR. يسمح هذا الإعداد باسترداد البيانات في حالة فشل قرص واحد، مما يجعل RAID-4 أكثر موثوقية من RAID-0 ولكنه أقل كفاءة في السيناريوهات التي تتطلب كتابة مكثفة بسبب الاعتماد على قرص تكافؤ مخصص.

في الشكل، يمكننا ملاحظة عمود واحد (قرص) مخصص للتكافؤ.

يتم حساب التكافؤ باستخدام دالة XOR بسيطة. إذا كانت بتات البيانات 0,0,0,1 فإن بت التكافؤ هو XOR(0,0,0,1) = 1. وإذا كانت بتات البيانات 0,1,1,0 فإن بت التكافؤ هو XOR(0,1,1,0) = 0. والنهج البسيط هو أن عددًا زوجيًا من الواحدات ينتج عنه التكافؤ 0، وعدد فردي من الواحدات ينتج عنه التكافؤ 1.

افترض أنه في الشكل أعلاه، تم فقد C3 بسبب فشل القرص. بعد ذلك، يمكننا إعادة حساب بت البيانات المخزن في C3 من خلال النظر في قيم جميع الأعمدة الأخرى وبت التكافؤ. وهذا يسمح لنا باستعادة البيانات المفقودة.

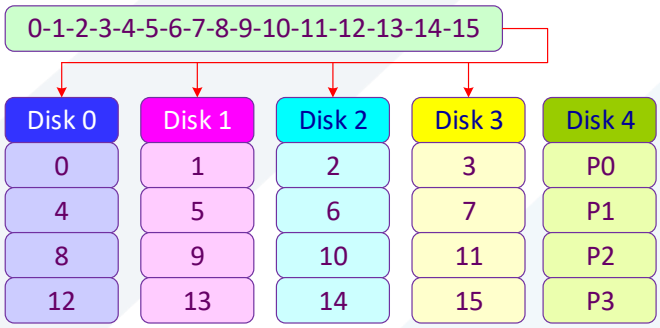
التقييم

الموثوقية:

يسمح RAID-4 باستعادة فشل قرص واحد على الأكثر (بسبب طريقة عمل التكافؤ). إذا فشل أكثر من قرص، فلا توجد طريقة لاستعادة البيانات.

السعة: (N-1)*B

يتم حجز قرص واحد في النظام لتخزين التكافؤ. وبالتالي، يتم توفير ((N-1قرص لتخزين البيانات، حيث يحتوي كل قرص على B كتلة.



المزايا

يساعد في إعادة بناء البيانات إذا فقدت بيانات واحدة على الأكثر.

العيوب

لا يمكنه المساعدة في إعادة بناء البيانات عندما يتم فقدان أكثر من واحدة.

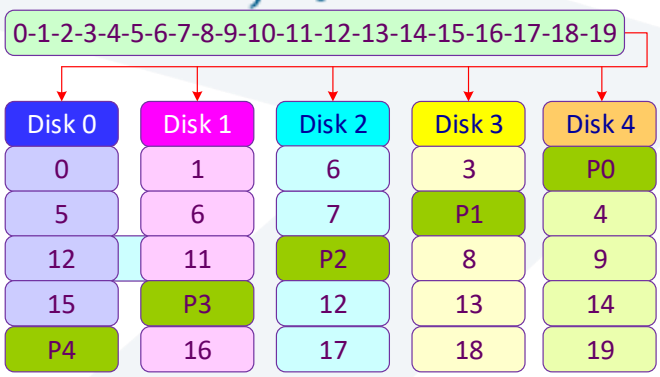
6. RAID-5 *(التجزئة على مستوى الكتلة مع التكافؤ الموزع)*

يعتمد RAID-5 على RAID-4 من خلال توزيع معلومات التكافؤ عبر جميع الأقراص بدلاً من تخزينها على محرك تكافؤ مخصص. يعمل هذا التكافؤ الموزع على تحسين أداء الكتابة بشكل كبير، وخاصة لعمليات الكتابة العشوائية، مع الحفاظ على تحمل الأخطاء لفشل القرص الفردي. يعد RAID-5 أحد أكثر تكوينات RAID استخدامًا نظرًا لتوازنه بين الموثوقية والأداء وكفاءة التخزين.

تقسيم البيانات على مستوى الكتلة: يتم تقسيم البيانات إلى كتل وتقسيمها عبر جميع محركات الأقراص في المصفوفة.

التكافؤ الموزع: يتم توزيع بتات التكافؤ، المحسوبة باستخدام دالة XOR، عبر جميع محركات الأقراص في نمط دوار. يضمن الدوران عدم تحميل أي قرص واحد بجميع عمليات التكافؤ، مما يقلل من الاختناقات.

استرداد البيانات: في حالة فشل قرص واحد، يمكن إعادة بناء البيانات المفقودة عن طريق إجراء عملية XOR على كتل البيانات المتبقية ومعلومات التكافؤ.



التقييم

الموثوقية:

يسمح RAID-5 باسترداد فشل قرص واحد على الأكثر (بسبب طريقة عمل التكافؤ). إذا فشل أكثر من قرص، فلا توجد طريقة لاسترداد البيانات. وهذا مماثل لـ RAID-4.

السعة: (N-1)*B

بشكل عام، يتم استخدام مساحة تعادل قرص واحد في تخزين التكافؤ. وبالتالي، يتم توفير أقراص ((N-1التخزين البيانات، حيث يحتوي كل قرص على كتل B.

المزايا

- يمكن إعادة بناء البيانات باستخدام بتات التكافؤ.
- يجعل الأداء أفضل.

العيوب

- تقنيتها معقدة وتتطلب مساحة إضافية.
- إذا تعرض كلا القرصين للتلف، فستضيع البيانات إلى الأبد.

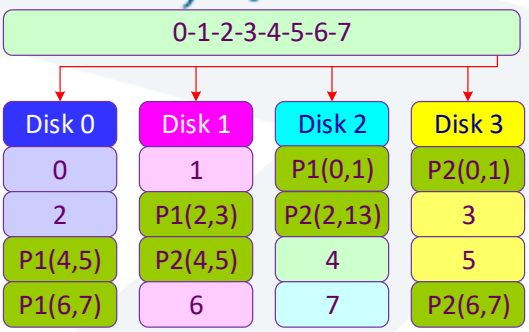
7. RAID-6 *(تجريد مستوى الكتلة باستخدام بتات تكافؤ)*

RAID-6 هو إصدار متقدم من RAID-5 يوفر تسامحًا محسنًا مع الأخطاء من خلال تقديم تكافؤ موزع مزدوج. وهذا يسمح لـ RAID-6 بالتعافي من فشل ما يصل إلى قرصين في وقت واحد، مما يجعله أكثر موثوقية للأنظمة الحرجة ذات المصفوفات الأكبر. ومع ذلك، يمكن أن تؤثر حسابات التكافؤ المضافة على أداء الكتابة.

التقسيم على مستوى الكتلة: يتم تقسيم البيانات إلى كتل وتقسيمها عبر جميع الأقراص في المصفوفة.

التكافؤ الموزع المزدوج: يتم حساب مجموعتين من معلومات التكافؤ لكل كتلة وتوزيعها عبر جميع الأقراص في المصفوفة بنمط دوار.

استرداد البيانات: إذا فشل قرص أو قرصان، يمكن إعادة بناء البيانات المفقودة باستخدام البيانات المتبقية ومعلومات التكافؤ.



التقييم

الموثوقية:

يمكن لـ RAID-6 تحمل الفشل المتزامن لقرصين، مما يوفر قدرة أكبر على تحمل الأخطاء من RAID-5.

السعة:

السعة القابلة للاستخدام = $(N-2)\times B$ حيث N هو العدد الإجمالي للأقراص وB هو عدد الكتل لكل قرص.

المزايا

- إمكانية وصول عالية جدًا للبيانات.
- معاملات قراءة البيانات بسرعة.

العيوب

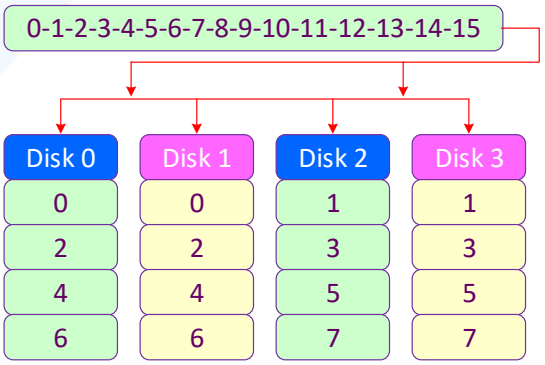
- نظرًا للتكافؤ المزدوج، فإن معاملات كتابة البيانات بطيئة.
- مطلوب مساحة إضافية.

RAID level 10 – combining RAID 1 & RAID 0

من الممكن الجمع بين مزايا وعيوب RAID 0 و RAID 1 في نظام واحد.

هذا هو تكوين RAID متداخل أو هجين.

يوفر الأمان من خلال عكس جميع البيانات على محركات الأقراص الثانوية مع استخدام التقسيم عبر كل مجموعة من محركات الأقراص لتسريع عمليات نقل البيانات.



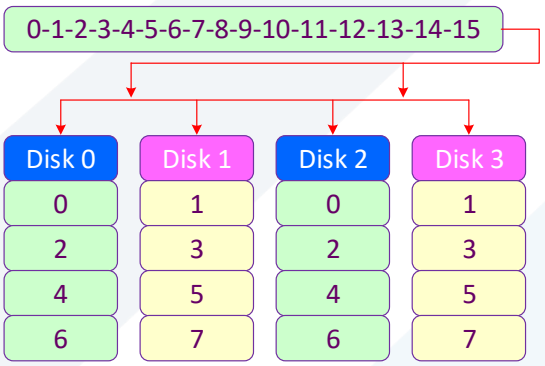
الشكل : RIAD 10

RAID level 01 – combining RAID 0 & RAID 1

من الممكن الجمع بين مزايا وعيوب RAID 0 و RAID 1 في نظام واحد.

هذا هو تكوين RAID متداخل أو هجين.

يوفر الأمان من خلال عكس جميع البيانات على محركات الأقراص الثانوية مع استخدام التقسيم عبر كل مجموعة من محركات الأقراص لتسريع عمليات نقل البيانات.



الشكل : RIAD 01