تطبيق محوسب لجدولة القرص ذي الرأس المتحرك بإتجاه المتطلب الأقرب

علي صاحب حمّود* / جامعة كربلاء / كلية الطب البيطري
E-mail : ali_abutaheen@yahoo.com

Keywords : Operating systems , disk , head , schedule .

خلاصة البحث

يسعى الباحثون في برمجيات أنظمة التشغيل Operating Systems إلى الحصول على هدفين أساسيين أولهما توفير مساحة خزن كافية بحيث تكون كفوءة وملاءمة للتطبيقات الحاسوبية المختلفة للمستفيدين في أجهزة الخزن المتنوعة, وثانيهما الوصول السريع إلى البيانات المخزونة في تلك الأجهزة من أجل حدوث عمليتي القراءة والكتابة.
ويعد القرص ذو الرأس المتحرك moving-head disk واحدا من أجهزة الخزن المهمة. هذا البحث يهدف إلى إيجاد طريقة جديدة تضاف إلى الطرق المعروفة في جدولته disk scheduling للوصول إلى البيانات.

Computerized application to disk scheduling for moving head disk to the closest requirement .

Abstract

The researchers in the operating systems are trying to got two objectives ;the first is securing the effective storage area , and suitable to the different applications. The second objective is the fast access to the data in the storage devices.
The moving-head disk one of the important storage devices. This research aimed to find new method to disk scheduling.

1 - المقدمة:

لقد انفرد ظهور الحاسوب في منتصف القرن ليكون الحدث الأهم والأكبر في تاريخ التقنية برمته، وبه بدأ عصر جديد عُرف بعصر (تقنية الحاسوب). وعالم التقنية لم يلق صدىً علمياً موازياً على مستوى التقنية على نحو ما لاقته تقنية الحاسوب. فمنذ ولادة أول حاسوب في أربعينات القرن العشرين، حدث تطور هائل في تقنية الحاسوب خلال النصف الثاني من القرن، في الأجهزة والبرمجيات، على حد سواء، فتحولت من حواسيب كانت تشغل مباني وغرفاً كاملة إلى حواسيب توضع على سطح المكتب وفي الحقائب.[1]

إن المعلوماتية بنيان أساسه المعرفة والمعلومات في شتى المجالات ومرافقه ثلاثة أنواع من التقنية: تقنية الحاسوب، وتقنية المعلومات، وتقنية الاتصالات، وقاعدته صناعة البرمجيات.

لمّا كان نظام التشغيل operating system وهو قلب البرمجيات software (البرمجيات هي كل الأشياء التي هي ليست ماديّة Hardware [2]) والذي هو مديرا للموارد resources manager ومسيطرا عليها فأنه بذلك يعتبر الجزء المهم في عمل الحاسوب ومقياسا لكفاءتها. فنظام التشغيل يهدف إلى توفير سهولة العمل للمستفيد، وأيضا الفعل الجيد والمناسب لنظام الحاسوب. [3]

يقوم نظام التشغيل بتقديم النموذج المناسب للمعالجات the processes model وإدارة الذاكرة memory management وإدارة عمليات الإدخال والإخراج I / O devices management ومن هذه الأجهزة سنحاول دراسة جدولة القرص ذو الرأس المتحرك ونبحث بطريقة جديدة لحركة الرأس في الوصول إلى البيانات المسجلة على هذا القرص. وهي حركته باتجاه الطلب الأقرب أولا ، ثم الاستمرار بخدمة المتطلبات requirements الأخرى على الاتجاه نفسه دون الأهتمام بقربه أو بعده من حركة الرأس.

إن القرص ذو الرأس المتحرك يملك شكلاً مدوراً ومسطحاً، وعندما يكون القرص في حالة التشغيل تكون السواقة drive بأعلى سرعة لتضع الرأس R / W head على سطح القرص. إن سطح القرص مقسم إلى مسالك tracks وهذه المسالك مقسمة إلى sectors، وأن البيانات التي تخزن على القرص تسجل مغناطيسيا في هذه المسالك بواسطة الرأس.

وهناك طرائق أربع لهذه الجدولة. [4]

1) الجدولة FCFS First – Come – First – Served

* ماجستير في علوم الحاسبات - مدرس مساعد في كلية الطب البيطري - جامعة كربلاء

ليكن ترتيب متطلبات الخدمة للمسالك هي كما وردت في الطابور queue و كما يأتي :
queue = 98، 183، 37، 122، 14، 124، 65، 67
وعلى فرض أن R / W head يبدأ من المسلك 53، إذن سيتحرك نحو المسلك 98 ثم المسلك 183 وهكذا حتى يصل المسلك 67 وبذلك يكون مجموع حركاته 640 مسلك. كما مبين بالشكل (1).
إن المشكلة هنا هو انتقال الرأس من المسلك 122 الى المسلك 14 ثم الى المسلك 124 مما زاد من مجموع حركات الرأس فزاد بذلك زمن المعالجة.

2) الجدولة SSTF Shortest – Seek – Time – First

الآن نعطي الخدمة لجميع المتطلبات القريبة لحركة الرأس، ثم الانتقال بعيدا إلى مكان آخر لإعطاء الخدمة لطلبات أخرى، إن هذه الطريقة تختار أقل ما يمكن من حركات الرأس ونتيجة لذلك تقلل من زمن المعالجة. لنرى الطابور نفسه 67، 65، 124، 14، 122، 37، 183، 98 = queue
فنلاحظ أن أقرب مسلك إلى المسلك 53 هو المسلك 65 ثم المسلك 67 وهكذا فيكون مجموع حركات الرأس 236 مسلكا وهي حوالي 1/3 الحركات في FCFS. وبذلك قل زمن المعالجة إلى الثلث. وكما هو واضح بالشكل (2).
إن المشكلة الرئيسة هنا أن متطلبات الخدمة للمسلك 183 ستتأخر مهما كانت هي مهمة لحين الانتهاء من المتطلبات القريبة. [5]
وهناك مشكلة أخرى وهي في حالة وجود طلبات للخدمة في وسط القرص حينها سيبقى الرأس يقدم خدماته في وسط القرص حتى لو كان هناك طلبا مهما للخدمة يقع بعيدا عن حركة الرأس. [6]

3) الجدولة SCAN

هنا يبدأ الرأس بالاتجاه نحو إحدى نهايتي القرص، ثم ينتقل بالاتجاه نحو النهاية الأخرى فيقدم خدماته لكل المسالك التي يمر بها و عندما يصل إليها يبدأ حركته نحو النهاية الأولى قائما بجميع وظائفه. فعليه فأن الرأس يقوم بمسح scan القرص من النهاية إلى النهاية الأخرى وبالعكس. لنرى الطابور نفسه
67، 65، 124، 14، 122، 37، 183، 98 = queue و كما هو واضح بالشكل (3).
هنا سيقوم الرأس بخدمة المتطلبات التي يمر بها بالرغم من وجود متطلبات أخرى سبقت طلب الخدمة ولكنها لم تخدم.

4) C – SCAN

هذه الطريقة تشبه SCAN لكن الفرق إنه عندما يتجه الرأس نحو إحدى النهايتين فأنه يرجع مباشرة إلى النهاية الأولى دون أن يخدم أي متطلبات أخرى أثناء رجوعه. لنرى الطابور نفسه
67، 65، 124، 14، 122، 37، 183، 98 = queue وكما مبين بالشكل (4).
إن أيّ من هذه الطرائق لا يمكن أن تكون مثالية، فالجميع لها محسنها ومساويئها، وقد تكون الجميع ذات كفاءة واحدة. [7]

2 – بيان المشكلة :

من دراسة الطريقتين FCFS و SSTF تظهر لنا مشكلة ذات أهمية وهي حينما تتلاشى الميزات الحسنة التي قد تتوفر في أي منها.
فلندرس المثال الآتي :
لدينا الرأس المتحرك ضمن 200 مسلك في القرص ذو الرأس المتحرك، إنّه يبدأ خدماته من المسلك 85
وقد كان طابور المتطلبات وفق الترتيب الآتي :
queue = 90، 79، 102، 55، 150
لننظر إلى الحل وفق الطريقة FCFS ولنلاحظ الشكل (5) فنرى أن الرأس تحرك من المسلك 85 إلى المسلك 90 وهكذا حسب ما ورد في الطابور، إن مجموع حركاته هي 181 مسلكا.
والآن ليكن مثالنا هو حسب طريقة SSTF، ولنلاحظ الشكل (6) فنرى أن الرأس قد تحرك من المسلك 85 إلى المسلك 90 إذ هو الأقرب إليه ثم المسلك الأقرب فالأقرب ونحسب حركات الرأس فنجدها 181 مسلكا ونطابق الشكلين (5) و(6) فنجدها متشابهة. وبذلك فقدت SSTF واحدة من أهم حسناتها.

أما بالنسبة للطريقتين الأخيرتين SCAN & C-SCAN فأنهما في كل الأحوال تقومان بمسح كامل للقرص من النهاية الأولى إلى النهاية الثانية وهكذا.
إن هاتين الطريقتين يصدفان المعالجة السريعة و هما عادة بعيدتان عنها لكونهما يمسحان القرص كله .

3 – معالجة المشكلة (خدمة الطلب الأقرب وتقديم بقية الخدمات بالإتجاه نفسه) :

نحن بصدد معالجة المثال الأخير ونقترح إجراء حركة الرأس لتقديم خدماته حسب ما يأتي :

- يخدم الرأس المتطلب الأقرب إليه من بداية حركته و الذي هو بحاجة إلى خدمة فيؤدي خدماته إليه.

فإذا إفترضنا أن بداية الخدمة بدأت بالمسلك N فان المسلك الأقرب قد يكون بعده N+1 أو قبله N-1.
و رمز المسلك N هنا أكبر من رمز المسلك N-1 و في الوقت نفسه هي أصغر من رمز المسلك N+1.

- الاستمرار بتقديم الخدمات إلى المتطلبات الأقرب فالأقرب والاتجاه نحو آخر المتطلبات والتي تقع على الاتجاه ذاته. أي يكون بعد المسلك N+1 المسلك الذي بعده بترتيب متطلبات الخدمة و ليكن N+2 . و على هذا يكون المسلك N-2 طالبا للخدمة بعد المسلك N-1.
- بعد إنجاز خدماته إلى المتطلبات الأخيرة بالأتجاه الأول يرجع الرأس ليخدم المتطلبات الأخرى بالاتجاه الآخر وأيضا الأقرب فالأقرب.كما هو مبين بالمخطط الأنسيابي في الشكل (7).
- ينتهي تقديم الخدمة للمسالك بإنتهاء المتطلبات التي في القرص.

4 – النتائج و المناقشة :

لنفحص المثال والطابور هو 150، 55، 102، 79، 90 = queue وبداية انطلاق الرأس هو المسلك 85. لاحظ الشكل (8).
لنحسب مجموع حركات الرأس فنجدها تساوي 160 مسلكا وهي بذلك أقل من مجموع حركاته بطريقة FCFS والطريقة الأخرى SSTF.
الآن لنرى ماذا يحدث بالنسبة للطابور في المثال (1) ولنعالج الطابور
67، 65، 124، 14، 122، 37، 183، 98 = queue وعلى فرض أنه ينطلق من المسلك 53.
لاحظ الشكل (9).
إن مجموع حركات الرأس هي 299 مسلكا، وهي أكثر بقليل من الطريقة SSTF .و أقل بكثير من FCFS.

5– الإستنتاجات :

1. إن طريقة الحل للمشكلة كانت جيدة بالرغم من ان سرعتها أقل من سرعةِ SSTFِ بقليل .
2. إن جميع هذه الطرائق لا يمكن أن تكون مثالية، فالجميع لها محاسنها ومساوئها، وقد تكون الجميع ذات كفاءة واحدة أحيانا. .
3 . ربّما يذهب الرأس بعيدا عن المتطلبات التي كانت قريبة منه ولم يستطع خدمتها إلاّ بعد رجوعه ، لكن هذا لم يقلل من نجاح الطريقة بحل المشكلة .
4. هذه الطريقة يمكنها الابتعاد عن خدمة المتطلبات داخل الأسطوانة الواحدة one cylinder وذلك لقيام الرأس بخدمة المتطلبات الأقرب بالاتجاه نفسه.

المصادر

1.محمد نعمان مراد، مفاتح القرن الواحد والعشرين: تقنية الحاسوب والصناعة البرمجية والمعلوماتية، مكتب جهينة للطباعة، بغداد،2001.

2. Glib , Tom. "Principles of software engineering management" . Addison- Wesley,1988.

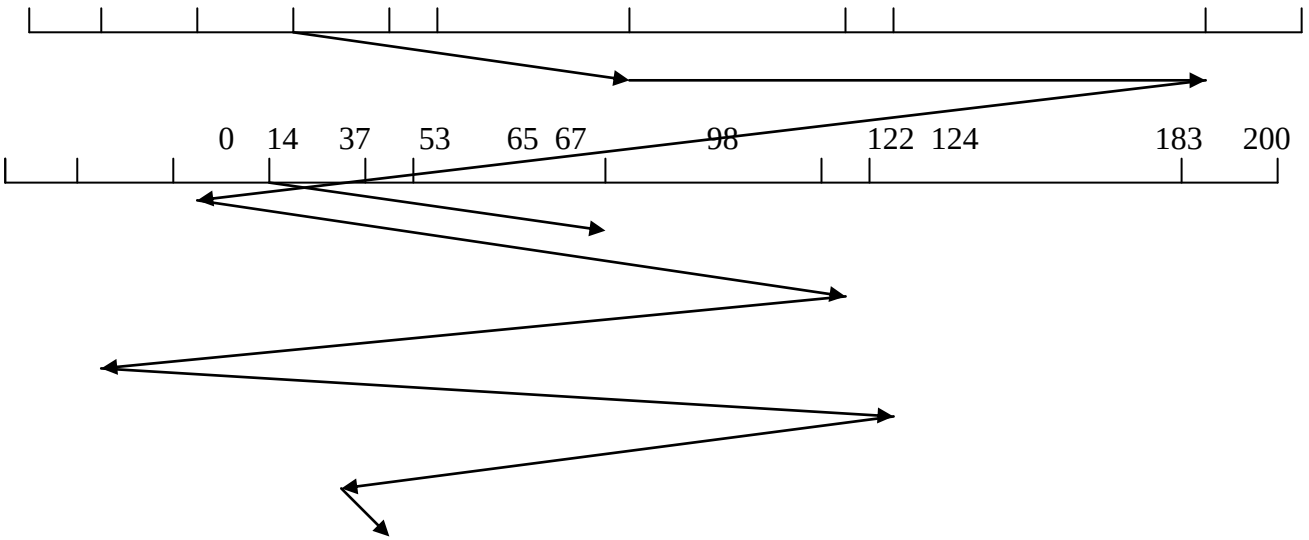
3.Peterson , J.L. and Silberschatz , A. "Operating sysetem concepts". Addison- Wesley Publishing Company. 2nd edition , 1985.

4. Tanenbaum , S. A. and Woodhull , S.A. "Operating systems design and implementation". Pearson prentice Hall , 3rd edition , 2006.

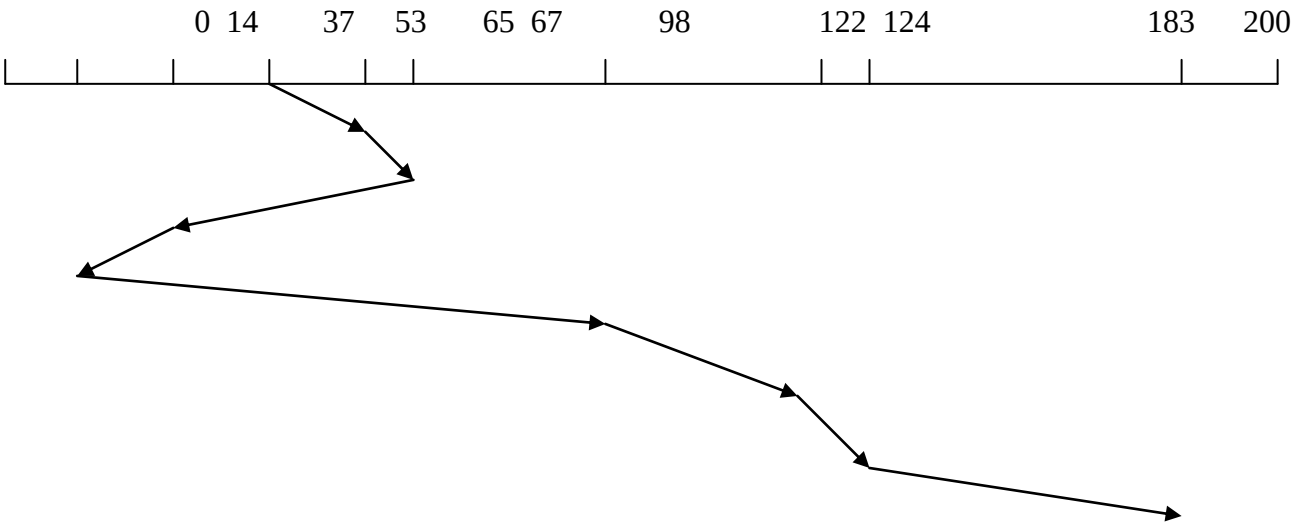
5. Tanenbaum , S. A. "Modern Operating Systems". Prentice-Hall International, Inc. 1992.

6.Stallings , W. "Operating systems internals and design principles". Pearson Education International , 5th edition , 2005.

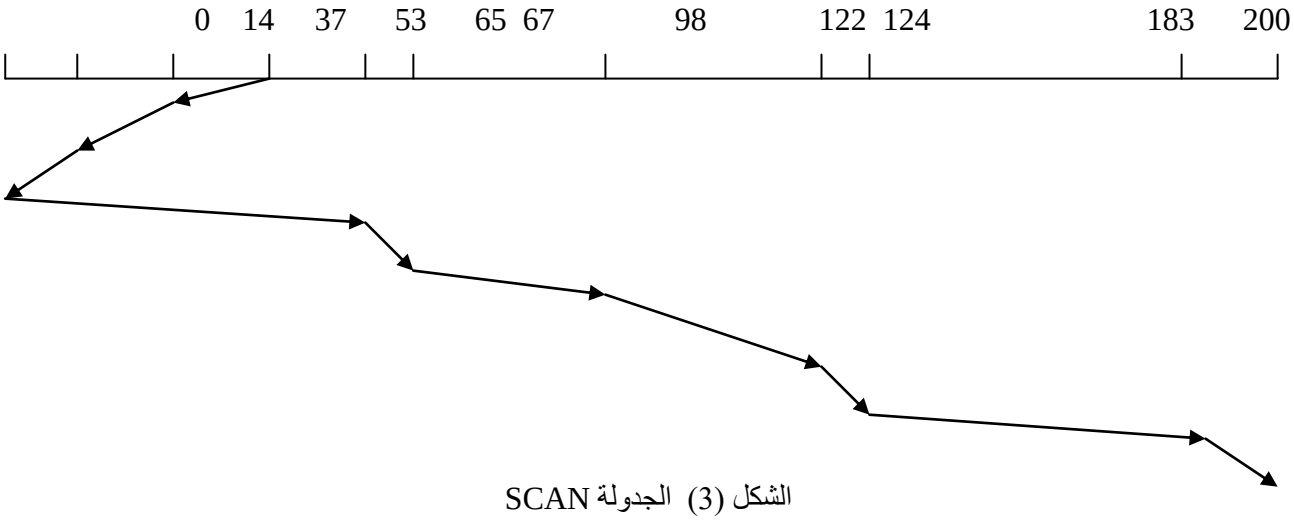
7. Silberschatz , A. and Galvin , P. and Gagne , G. " Applied operating system concepts". John Wiley & Sons, Inc. 1st edition , 2000.



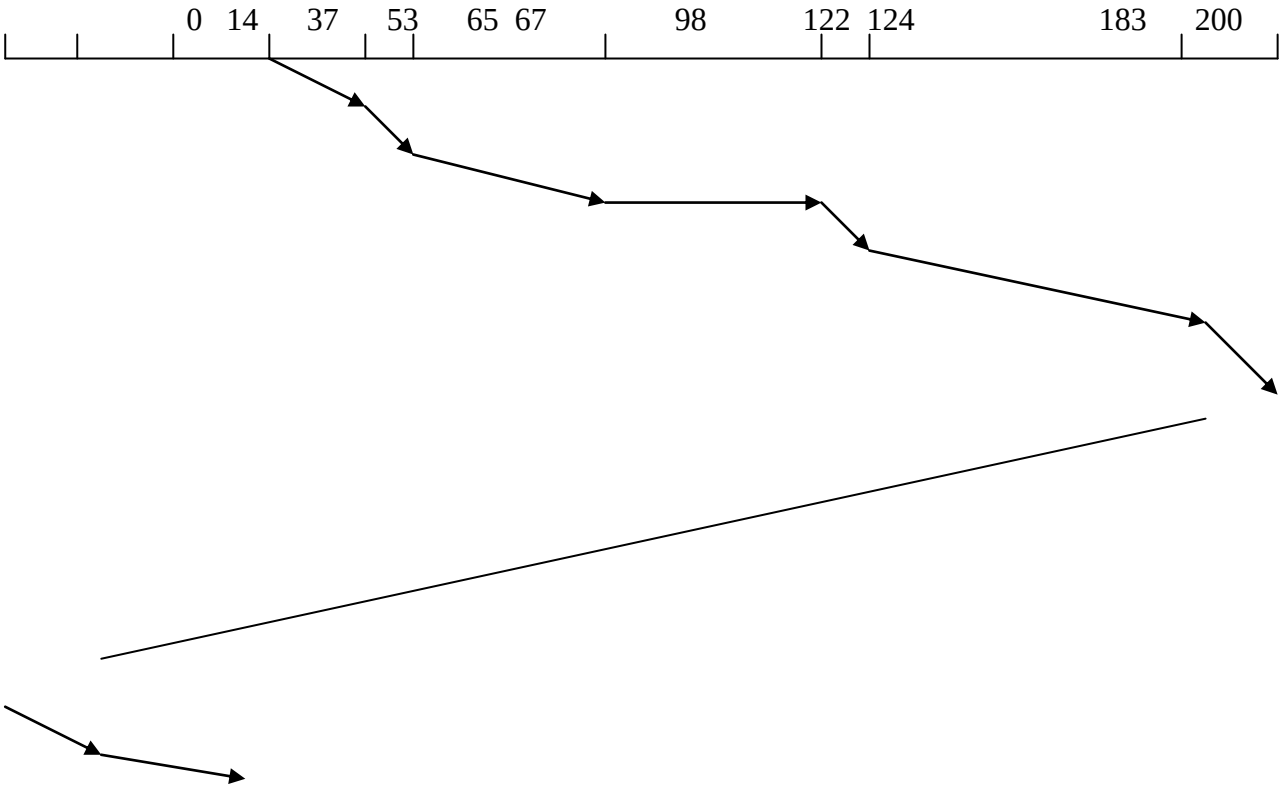
الشكل (1) الجدولةFCFS



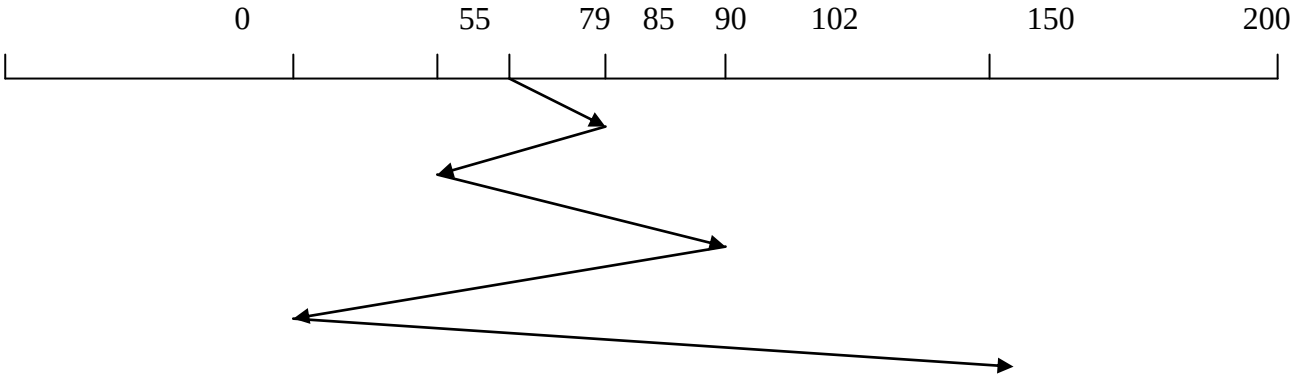
الشكل (2) الجدولة SSTF



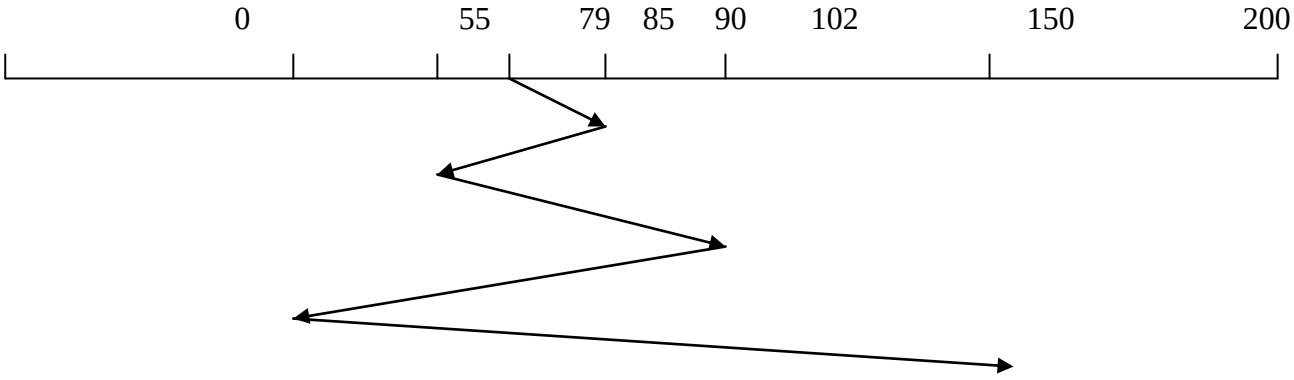
الشكل (3) الجدولة SCAN



الشكل (4) الجدولة C – SCAN

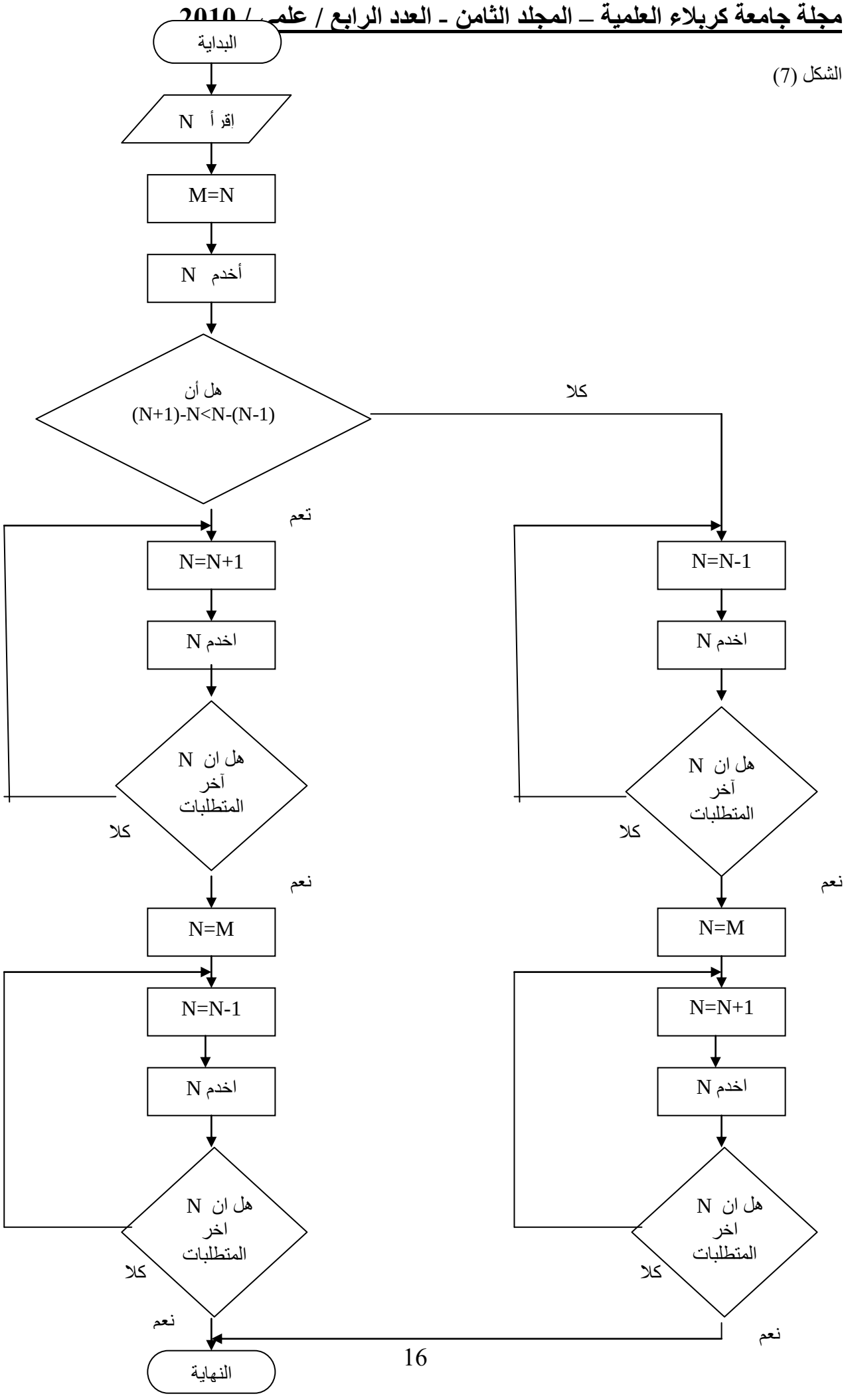


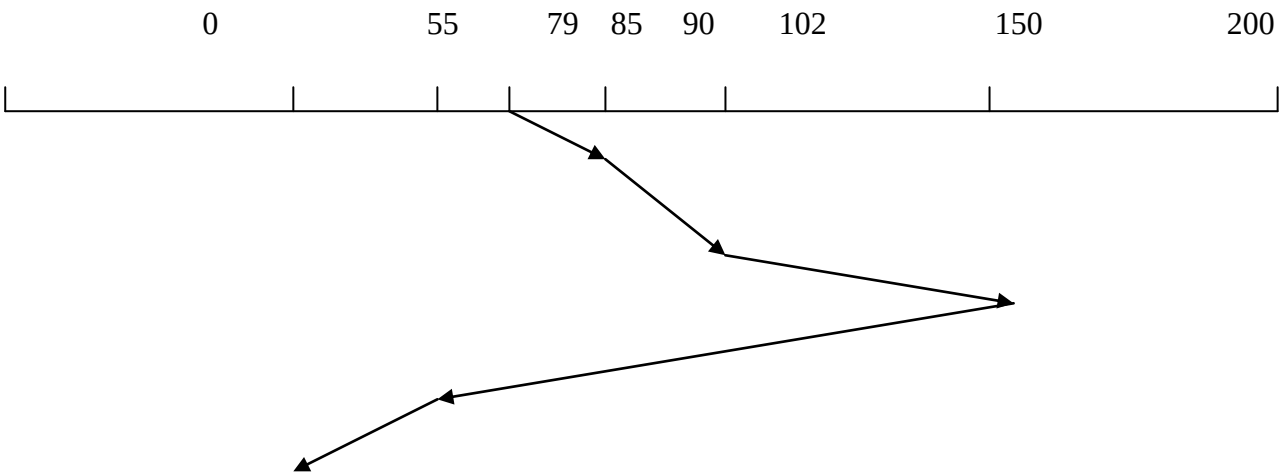
الشكل (5) يبين مشكلة الجدولة FCFS



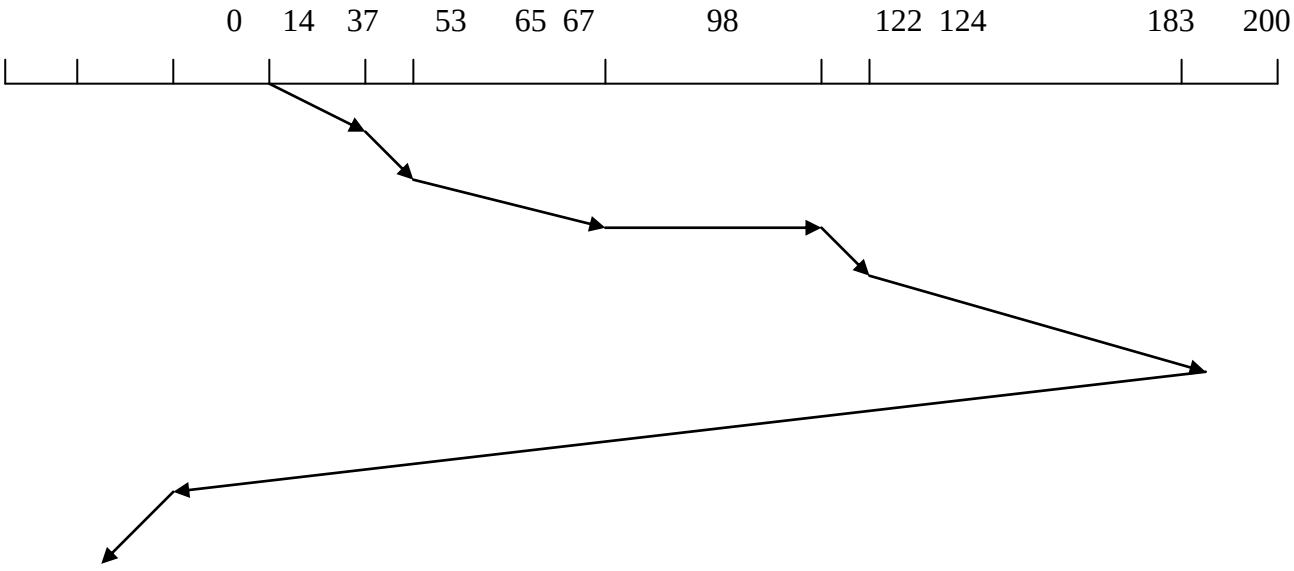
الشكل (6) يبين مشكلة الجدولة SSTF

الشكل (7)


البداية
إقرأ N
M=N
أخدم N
هل أن
(N+1)-N<N-(N-1)
كلا
نعم
N=N+1
اخدم N
هل ان N
آخر
المتطلبات
كلا
نعم
N=M
N=N-1
اخدم N
هل ان N
اخر
المتطلبات
كلا
نعم
N=N-1
اخدم N
هل ان N
آخر
المتطلبات
كلا
نعم
N=M
N=N+1
اخدم N
هل ان N
اخر
المتطلبات
كلا
نعم
النهاية



الشكل (8) يبين مناقشة طريقة البحث



الشكل (9) يبين طريقة الجدولة في البحث