المجلة العراقية للعلوم الاحصائية (14) 2008
ص ص [56–67]

خوارزمية جدولة مقترحة للمعالجات المتعددة

أحمد محمود السبعاوي* **غزوان هاني محمود****

الملخص

يتضمن هذا البحث عرض خوارزمية جديدة لجدولة المعالجات المتعددة في الحاسبات الإلكترونية. وقد عرضت مشكلة الجدولة بافتراض أن لدينا مجموعة من العمليات (n) تمثل بواسطة الرسم، مطلوب تنفيـذها علـى عـدد اختيـاري مـن المعالجات المستقلة (m)، وكل عملية تتطلب (τ) من الوقت لتنفيذها، وتمت مقارنة الخوارزمية المقترحة مع خوارزمية جدولـة (CP/MISF) باسـتخدام المحاكـاة (Simulation) وأظهرت كفاءة عالية وسهولة بالاستخدام.

New Scheduling Algorithm for Multiprocessor

ABSTRACT

This research demonstrates a new scheduling algorithm for multiprocessor in computer system, the problem of scheduling were dealt in the case of (n) jobs which are represented by graph, and the aim is to apply it on (m) independent processor, each job requires (τ) running time. This study finds out a new algorithm for scheduling a set of job for an arbitrary number of identical processors. This algorithm was compared with scheduling algorithm (CP/MISF) by using simulation and showed high efficiency and easy to use.

---

* استاذ مساعد/ قسم الاحصاء والمعلوماتية/ كلية علوم الحاسبات والرياضيات

** مدرس مساعد/ كلية علوم الحاسبات والرياضيات

تاريخ التسلم : 19 /2007/3 ـــــــــــــــــــــــ تاريخ القبول : 21 /2007/8

1. المقدمة

تعد إدارة العمليات داخل المعالجات من أهم أعمال المحاسبة الإلكترونية، وتظهر هذه العمليات إلى حيز الوجود عند القيام بتشغيل أحد الأوامر أو البرامج، في هذه الحالة تعطى الحاسبة الإلكترونية كل عملية في حيز التشغيل رقماً، وتقوم وحدة المعالجة المركزية CPU (Central Processor Unit) بإنجاز هذه العمليات واحدة تلو الأخرى، وذلك بتقسيم الوقت ما بين هذه العمليات وخدمة كل عملية حسب الضرورة والمرتبة الموضوعة لها.

أدت التطورات المعقدة في الحاسبات إلى ظهور الحاسبات ذات المعالجات المتعددة التي تستطيع أن تنفذ عدة عمليات في الوقت نفسه، مما أدى إلى ظهور مشكلة جدولة هذه المعالجـات، (Multiprocessor Scheduling Problem)، وذلك لإدارة هذه العمليات داخل المعالجات بحيث تمكن الحاسبة الإلكترونية من إنهاء العمل على المعالجات بأقل (أمثل) وقت ممكن.

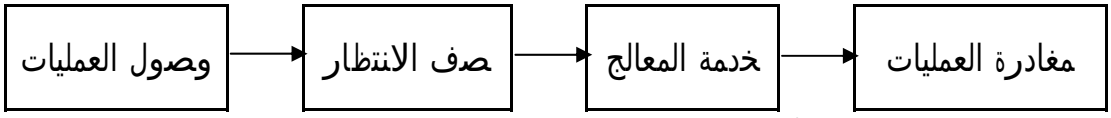
لهذا اتجه عدد من الباحثين الى دراسة مشكلة الجدولة على نظـام الحاسـبات الإلكترونيـة ذات المعـالج الواحـد وذات المعالجـات المتعـددة وقـد درس [kwok, Ahmed(1999)] مشكلة الجدولة ووضحا ان خوارزميـات الجدولـة تستخدم لتقليل طول الجدولة في المعالجات المتعددة وذلك لايجـاد امثـل جدولـة وتناولا في البحث 27 خوارزمية جدولة متنوعة . وقـدم سـنة [keqin, (2000) yi]خوارزمية لجدولة مجموعة من العمليات مع عدم امكانية القطع للعمليات علـى عدد اختياري من المعالجات .

ففي أنظمة الحاسبات الإلكترونية ذات المعالجـات المتعـددة هنـاك عـدة عمليات تنفذ في الوقت نفسه. السؤال: ما النظام المستخدم لسحب عملية معينة مـن بين العمليات المنتظرة في صفوف الانتظار وتخصيص المعالج لها؟ بحيث تمكـن المعالجات من إنهاء العمل بأقل وقت ممكن.

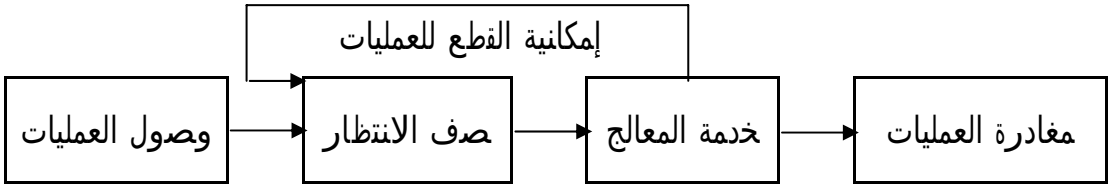
يهدف هذا البحث إلى إيجاد خوارزمية لجدولة مجموعة من العمليات على عدد اختياري من المعالجات المستقلة.

2. أنواع الجدولة

هناك نوعان من خوارزميات الجدولة، الأول هو خوارزميات الجدولة مـع عدم إمكانية القطع للعمليـات (Non Preemptive Scheduling Algorithm) في هذا النوع من خوارزميات الجدولة لا يمكن للعمليات أن تتحول من حالة التنفيذ إلى حالة الانتظار حتى تنفذ بأكملها، أما النوع الثاني من خوارزميات الجدولة فهو خوارزميات الجدولة مع إمكانية القطـع للعمليـات (Preemptive Scheduling Algorithm) في هذا النوع من خوارزميات الجدولة يمكن للعمليات أن تتحول من حالة التنفيذ إلى حالة الانتظار قبل إكمالها. وكما موضح بالأشكال الآتية:



أ. جدولة مع عدم إمكانية القطع للعمليات

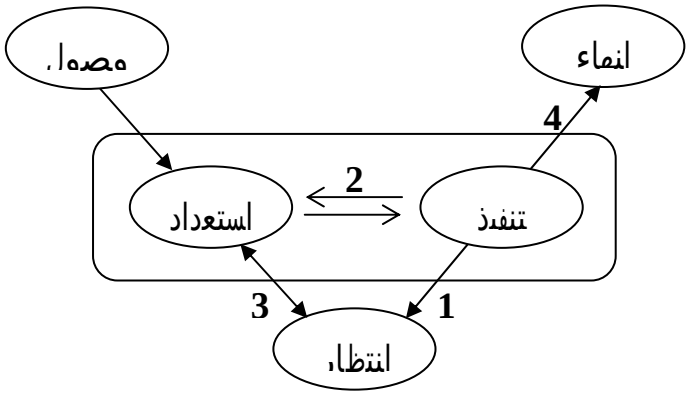


ب. جدولة مع إمكانية القطع للعمليات

الشكل (1): أنواع الجدولة

وبناءً على ما تقدم فإن قرارات الجدولة ربما تحدث تحت الحالات الأربع الآتية:

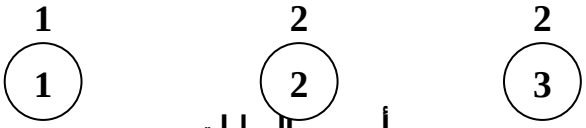
1. عندما تتغير العملية من حالة التنفيذ إلى حالة الانتظار.
2. عندما تتغير العملية من حالة التنفيذ إلى حالة الاستعداد.
3. عندما تتغير العملية من حالة الانتظار إلى حالة الاستعداد.
4. عندما تنتهي العملية (تنفذ بأكملها).



الشكل (2): مخطط انتقال العمليات

عندما تحدث الجدولة تحت الحالتين 1، 4 تسمى بالجدولة مع عـدم إمكانيـة القطع للعمليات (Non Preemptive Scheduling)، وعندما تحدث تحـت كـل الحالات الأربع تسمى بالجدولة مـع إمكانيـة القطـع للعمليـات (Preemptive Scheduling). هناك أسباب مختلفة لاستخدام خوارزميات الجدولة مـع إمكانيـة القطع للعمليات، وتتضمن سرعة الاستجابة ذات الأسبقية العالية، وغالباً ما تكـون إمكانية إنهاء العمل على المعالجات في حالة الجدولة مع إمكانية القطـع للعمليـات أسرع من إنهاء العمل على المعالجات في حالة الجدولة مع عدم إمكانيـة القطـع للعمليات. لذلك تعد خوارزميات الجدولة مع إمكانية القطع للعمليـات أفضـل مـن خوارزميات الجدولة مع عدم إمكانية القطع للعمليات ولكنها أكثر تعقيداً.

ولتوضيح الفرق بين نوعي الجدولة نعرض المثال الآتي:



أ. رسم العمليات

P_1	1	3	2
P_2	2	1	3

1 2 3

ب. جدولة مع عدم إمكانية القطع للعمليات

P_1	1	1	3	3
P_2	2	2	Φ	Φ

1 2 3 4

ج. جدولـة مـع إمكانيـة القطع للعمليات

مثال (1): الفرق بين أنواع الجدولة

3. معايير الجدولة

إن خوارزميات جدولة المعالجات لها خواص مختلفة، وربما تكون إحـدى هذه الخوارزميات أفضل من الأخرى، ولمعرفة كفاءة هذه الخوارزميات في تنفيـذ مجموعة من العمليات، لابد من معرفة أهم الأسس المتعارف عليها للمقارنـة بـين كفاءة خوارزميات الجدولة التي تمكننا من الحصول على الجدولـة المثلـى، ففـي أنظمة الحاسبات ذات المعالج الواحد فإن كفاءة خوارزميات الجدولة تعتمد على مـا يأتي:

1. العدالة (Fairness):

تعني العدالة أن كل عملية سوف تحصل على فرصة للتنفيـذ وذلـك بتخصـيص المعالج لها.

2. منفعة المعالج (CPU Utilization):

هو نسبة عدد الوحدات التي كان المعالج مشغولاً بتنفيذ العمليات على عدد الوحدات الكلي.

3. إنتاجية المعالج (CPU Throughput):

هو نسبة عدد العمليات المنفذة في النظام خلال وحدة الزمن.

4. وقت العودة (Turnaround Time):

هي الفترة الزمنية لتنفيذ العملية بأكملها، أو هي الفترة الزمنية بين وقـت وصـول العملية إلى وقت إنهائها.

5. وقت الانتظار (Waiting Time):

هي الفترة الزمنية التي تستغرقها العملية في صف الانتظار لكـي يخصـص لهـا المعالج.

6. وقت الاستجابة (Response Time):

هي الفترة الزمنية بين وقت وصول العملية إلى وقت بدء التنفيذ.

وفي أنظمة الحاسبات الإلكترونية ذات المعالجات المتعـددة، فضـلاً عـن المقاييس السابقة يمكن إضافة مقياس آخر هو:

7. طول الجدولة SL (Scheduling Length):

هي الفترة الزمنية التي تستغرقها جميع العمليات لتنفيذها، ففي حالة المعالجات المتعددة تكون أمثلية الجدولة معتمدة على طول الجدولة.

إن معظم خوارزميات الجدولة تعمل على زيادة منفعة المعالجات وإنتاجيتها وتقليل وقت العودة ووقت الانتظار ووقت الاستجابة وطول الجدولة وكذلك تحقيق العدالة بين العمليات، والأمثلية هي تحقيق الاتزان في كل مقياس من هذه المقاييس

4. الخوارزمية المقترحة

سيتناول هذا المبحث عرض خوارزمية جديدة لجدولة مجموعة من العمليات على عدد اختياري من المعالجات في الحاسبة الإلكترونية، وتعد هذه الخوارزمية غير تقليدية بالنسبة الى الخوارزميات الأخرى، اذ يمكن استخدامها إذا كانت العمليات مستقلة أو مرتبطة، وكذلك يمكن استخدامها إذا كانت أوقات التنفيذ للعمليات متساوية أو مختلفة. وعليه يمكن تقسيم هذه الخوارزمية على ثلاث خوارزميات فرعية وكما يأتي:

1. خوارزمية جدولة العمليات المرتبطة ذات أوقات التنفيذ المختلفة.
2. خوارزمية جدولة العمليات المرتبطة ذات أوقات التنفيذ المتساوية.
3. خوارزمية جدولة العمليات المستقلة.

وتعتمد هذه الخوارزمية على أوقات التنفيذ وكذلك على المستويات (Level).

إن طريقة عمل الخوارزميات الثلاث في أعلاه لها نفس الخطوات ويمكن توضيح ذلك وكما يأتي:

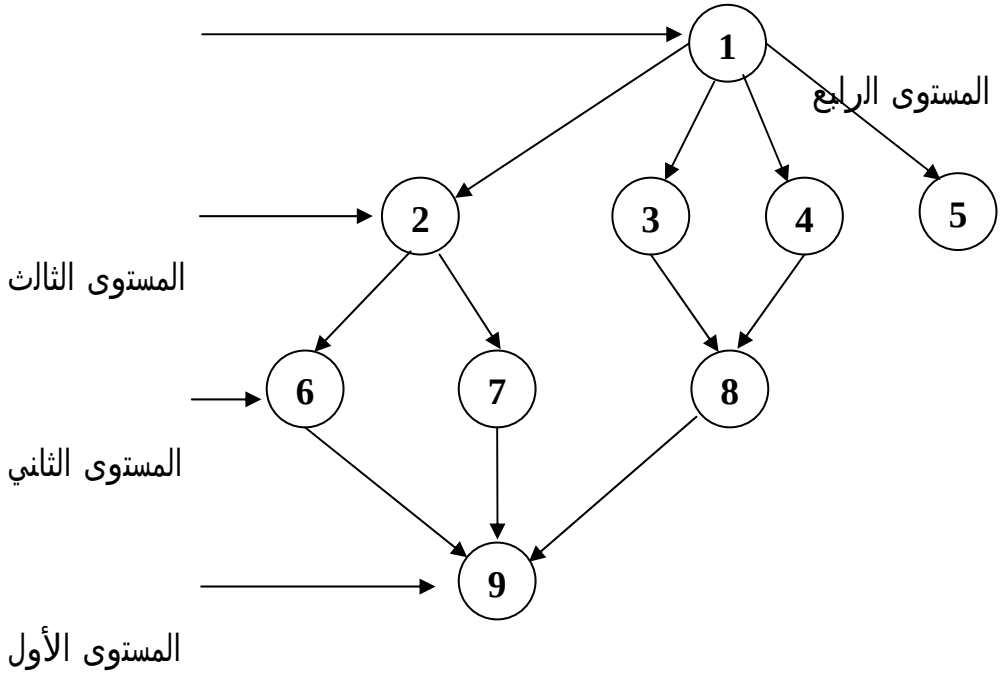
إذا كان لدينا مجموعة من العمليات المرتبطة ذات أوقات التنفيذ المتساوية أو المختلفة فإن المعالجات تبدأ بتنفيذ العمليات في المستوى الأخير (المستوى الأبعد عن مستوى عقدة النهاية) وتنفذ العمليات التي لها أعلى وقت تنفيذ من بين العمليات مع إمكانية القطع للعمليات (Preemptive). وبعد الانتهاء من تنفيذ العمليات في المستوى الأخير، ينتقل التنفيذ إلى المستوى الذي يليه.

أما إذا كانت العمليات مستقلة، فإن هذه العمليات تكون في مستوى واحـد، والخوارزمية تنفذ العمليات التي لها أعلى وقت تنفيذ من بين العمليات مع إمكانيـة القطع للعمليات.

خطوات الخوارزمية:

1. انتظار المعالج ليصبح عاطلاً عن العمل (السكون).
2. تعريف المجموعة R وهي كل العمليات التي تكون أسـلافها (Precedence) منفذة.
3. تعريف المجموعة R´ من R وهي كل العمليات التي تكون في أعلى مسـتوى (الأبعد عن مستوى عقدة النهاية).
4. اختيار العملية التي لها أكبر وقت تنفيذ من المجموعة R´ وتخصيص المعـالج لها وتنفيذ وحدة واحدة منها (Quantum).
5. إذا لا توجد عمليات أذهب إلى 1.

ولتوضيح طريقة عمل الخوارزمية الجديدة لنفرض أن لدينا العمليات في المثال(2).



أ. رسم العمليات

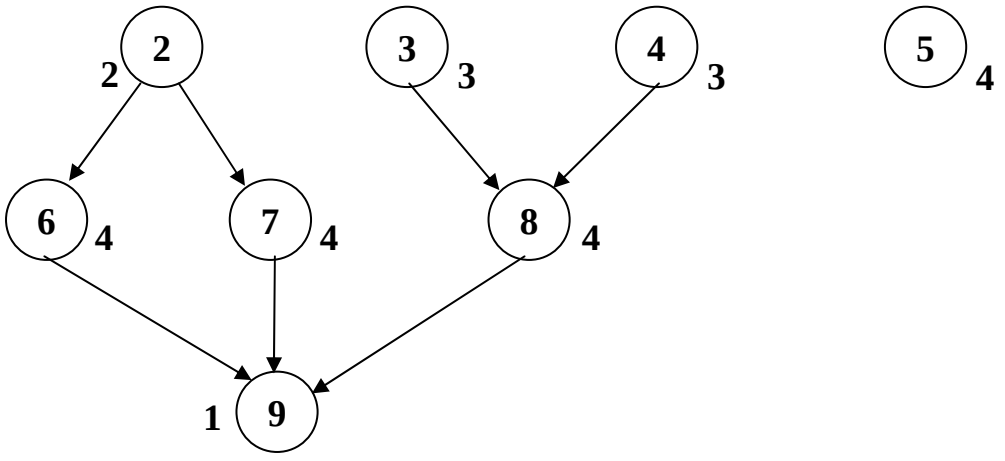
P_1	1	1	5	5	5	5	5	6	6	6	6	9
P_2	Φ	Φ	4	3	2	4	3	7	7	7	7	Φ
P_3	Φ	Φ	2	4	3	2	4	8	8	8	8	Φ
	1	2	3	4	5	6	7	8	9	10	11	12

ب. جدولة العمليات

الشكل (3): جدولة العمليات باستخدام الخوارزمية المقترحة

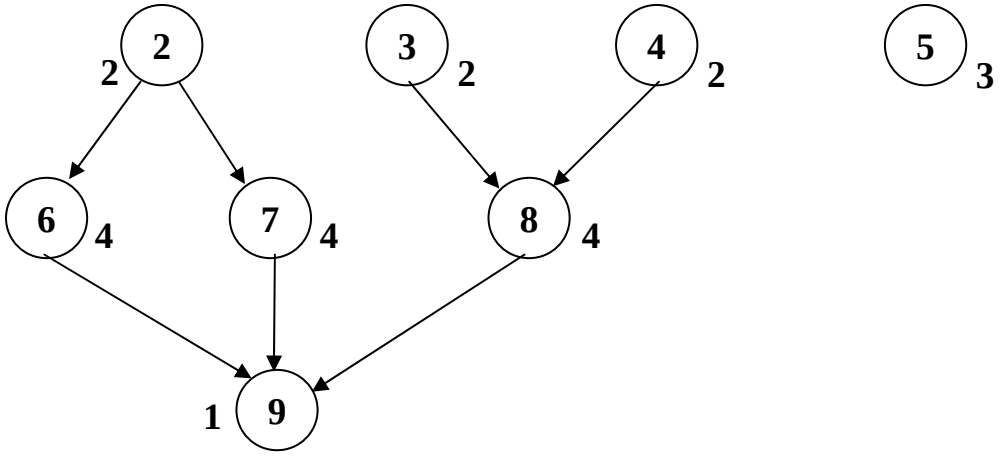
يبدأ التنفيذ بتخصيص المعالج الأول للعملية (1) لكونها من المستوى الرابع وليس لها أسلاف، ويبقى المعالجان الثاني والثالث عاطلين عن العمل، لأنه لا يمكن تنفيذ أية عملية أخرى بسبب اعتمادها على العملية الأولى.

العمليات (2, 3, 4, 5) جميعها من المستوى الثالث وأسلافها منفـذة (العمليـة 1)، سوف نختار العمليات التي لها أكبر وقت تنفيذ وهمـا العمليتـان (4, 5) أمـا العمليتان (2,3) فلهما نفس وقت التنفيذ وعليه سوف نختار العمليـة (2) لأن لهـا أخلافاً أكثر من العملية (3). ويخصص المعالج الأول للعمليـة (5) وينفـذ وحـدة واحدة منها (Quantum). ويخصص المعالج الثاني للعملية (4) وينفذ وحدة واحدة منها. ويخصص المعالج الثالث للعملية (2) وينفذ وحدة واحدة منها وبذلك يصـبح لدينا الشكل الآتي:



الشكل (4): توضيح طريقة عمل الخوارزمية المقترحة

لدينا العمليات (5,4,3,2) أسلافها منفذة ومن المستوى الثالث، نختار العمليات التي لها أكبر وقت تنفيذ وهي العمليات (5,4,3)، ويخصص المعالج الأول للعملية (5) وينفذ وحدة واحدة منها، ويخصص المعالج الثاني للعملية (3) وينفذ وحدة واحدة منها، ويخصص المعالج الثالث للعملية (4) وينفذ وحدة واحدة منها، وبذلك يكون لدينا الشكل الآتي:



الشكل (5): توضيح طريقة عمل الخوارزمية المقترحة

وهكذا إلى أن يتم تنفيذ جميع العمليات في المستوى الثالث ثم يتم الانتقال إلى المستوى الثاني، ويكون لدينا العمليات (8,7,6) أسلافها منفذة وجميع أوقات التنفيذ متساوية تخصص المعالج الأول للعملية (6)، والمعالج الثاني للعملية (7)، المعالج الثالث للعملية (8)، وهكذا إلى أن تنفذ جميع العمليات ثم تنفذ العملية (9) في المستوى الأول.

5. المحاكاة

نظراً لصعوبة الحصول على البيانات والمشاهدات المناسبة والدقيقة (لعدم توافرها)، فقد تم استخدام البرنامج الجاهز (minitab) لتوليد المشاهدات العشوائية اللازمة (72) حالة محاكاة لمعرفة كفاءة الخوارزمية المقترحة.

بعد توليد البيانات وتنفيذها عن طريق الخوارزمية المقترحة لتحديد طول الجدولة، نجري المقارنة بين الخوارزمية المقترحة وخوارزمية جدولة (CP/MISF) ويعتمد بشكل أساسي على الفرق بين طول الجدولة بين

الخوارزميتين. ومن المقارنة بين طول الجدولة للخوارزمية المقترحة مع خوارزمية الجدولة (CP/MISF) وجد أن (23) حالة من (72) حالة من حالات المحاكاة قـد انخفض طول الجدولة فيها كما مبين في الجدول الآتي:

الجدول (1)

عدد العمليات	المستويات	طول الجدولة باستخدام الخوارزمية المقترحة	طول الجدولة باستخدام خوارزمية CP/MISF
5	3	9	10
7	2	10	11
7	3	11	12
7	4	12	13
10	3	15	16
10	3	15	16
10	4	16	17
10	4	17	18
10	5	16	17
10	5	17	19
16	4	16	17
16	4	17	18
16	5	18	19
20	4	20	21
20	4	20	21
20	5	20	21
20	5	21	22
20	6	21	22
20	6	23	25
24	4	24	25
24	5	24	25
24	6	24	25
24	6	24	26

6. الاستنتاجات

1. توصلت الدراسة إلى خوارزمية ذات كفاءة لجدولة مجموعة من العمليات على عدد اختياري من المعالجات المستقلة، ويمكن تطبيق هذه الخوارزمية في حالة العمليات المرتبطة أو المستقلة، وكذلك إذا كانت أوقات التنفيذ للعمليات متساوية أو مختلفة.
2. نلاحظ أن طول الجدولة في بعض الحالات قد انخفض (تقليل طول الجدولـة) باستخدام الخوارزمية الجديدة مقارنة بخوارزمية جدولة (CP/MISF).
3. نلاحظ أن سرعة الاستجابة للعمليات باستخدام الخوارزمية الجديدة أسرع منـه باستخدام خوارزمية جدولة (CP/MISF).
4. إن زيادة العمليات تؤدي إلى زيادة كفاءة الخوارزمية المقترحة.

7. المصادر

1. الجبوري، غزوان هاني محمود (2002)،" **خوارزميات جدولة المعالج الواحـد والمعالجات المتعددة مع المحاكاة**"، رسالة ماجستير، كليـة علـوم الحاسـبات والرياضيات، جامعة الموصل، العراق، (غير منشورة).
2. Abraham, S. and Peter Baer, G. (1998), "**Operating System Concepts**", Addison-Wesley Publishing Company.
3. Adam, T. L., Chandy, K. M. and Dikson, J. R. (1974), "**A Comparison of List Scheduling for Parallel Processing System**", Commun. ACM, Vol. (17), No. (12), PP. 685-690.
4. Gonzalez, M. J. (1977), "**Deterministic Processor Scheduling**", J. ACM Comput. Survey, Vol. (9), No. (3), PP. 173-204.
5. Hu, T. C. (1961), "**Parallel Sequencing and Assembly Line Problems**", Oper. Res. Vol. (19), No. (6), PP. 841-848.

6. Hwang, K. and Briggs, F. A. (1984), "**Computer Architecture and Parallel Processing**", McGraw-Hill, Inc. New York.
7. Kasahara, H. and Narita, S. (1984), "**Ractical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing**", IEEE Trans. Comput., Vol. (C-33), No. (11), PP. 1023-1029.
8. Kwok, Y-K and Ahmed, I. (1999), "**Static Scheduling Algorithm for Allocating Directed Task Graphs to Multiprocessor**", J. ACM. Comput. Surv. Vol. (31), No. (4), PP. 406-471.
9. Nutt, G. J. (1992), "**Centralized and Distributed Operating System**", Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
10. keqin, l. and yi,pan. (2000), "**probabilistic analysis of Scheduling precedence constrained parallel tasks a Multiprocessor with contiguous processor allocation**", IEEE Trans. Comput. Vol. (49), No. (10), PP. 137-146.
11. Toepfer, J. A. (2000), "**When One Brain just is not enough! The Scheduling of Multiprocessor System**", Internet.