خوارزمية هجينة لجدولة المعالج باستخدام شريحة زمنية متغيرة*

نجم عبد الله عبد القادر الطحّان

مدرس مساعد، المعهد التقني، الموصل

المستخلص

تعدّ خوارزمية راوند روبن (Round Robin) من خوارزميات الجدولة التي تَستخدم شريحة زمنية ثابتة (static) خلال مدة التنفيذ، وفي المقابل فهي تعتمد على خدمة القادم أولاً يُخدم أولاً(First come first serve). صُممت خصيصاً لأنظمة تشغيل المشاركة الزمنية (Time sharing os). الهدف الرئيس لهذه الدراسة هو تطوير أسلوب جديد أو خوارزمية هجينة مقترحة لخوارزمية راوند روبن لجدولة المعالج المسندة بالأفضلية باستخدام شريحة زمنية متغيرة (dynamic). الشريحة الزمنية المتغيرة هي لتحسين أداء خوارزمية راوند روبن وتحسين إخفاقاتها في عدد خزن السياقات ومعدل زمن الانتظار ومعدل الوقت الدوري التي عادة ما تُثقِل النظام.

في هذه الدراسة صُممت الخوارزمية المقترحة لِجَمع محاسن خوارزميتي (راوند روبن والأفضلية). وقد أثبتت النتائج أن الخوارزمية المقترحة أفضل أداءً من الخوارزمية البسيطة (راوند روبن والأفضلية).

إذ إن جوانب التحسين يشمل معدل الوقت الدوري، معدل زمن الانتظار وعدد حالات خزن السياقات. أما نسبة التحسين تتفاوت بين الحالات الدراسية المعتمدة وطبيعة الحالة الدراسية في البحث، وعلى العموم يمكن ملاحظة الفرق بين أداء الخوارزمية البسيطة والخوارزمية المقترحة من خلال الجداول في نهاية كل حالة دراسية. تم استخدام برنامج Matlab لكفاءته العالية وخاصة في المصفوفات.

الكلمات المفتاحية: جدولة المعالج، خوارزمية راوند روبن، خوارزمية الأفضلية، الوقت الدوري، زمن الانتظار، شريحة زمنية، خزن السياقات.

(*) تم استلام البحث في ٢٠١٣/٩/٣ وحصل على قبول النشر في ٢٠١٤/٣/١٦.

Hybrid Algorithm for CPU Scheduling By Using Dynamic Time Quantum

Najim A. Abd-Alkader Al-Tahhan
Asst. Lect., Technical Institute, Mosul

Abstract

Round Robin (RR) is a kind of process scheduling algorithms, where static time quantum is used along the process execution. In the other hand it depends on the First Come First Serve (FCFS) algorithm. RR is designed especially for time sharing operating systems.

The primary objective of this paper is to develop a new approach or proposed hybrid priority based round robin CPU scheduling using dynamic time quantum. Dynamic time quantum is used to improve the performance of RR and performs once of the degrads with respect to Context Switches (CS), Average Waiting Time (Avg (WT)) and Average Turnaround Time (Avg (TAT)) that an overhead on the system.

In this paper, the proposed algorithm is designed to meet the advantage of RR algorithm and priority algorithm. The experimental results show that the proposed algorithm performs better than both RR, priority algorithms. Matlab has been used for its high efficiency particularly in matrixes.

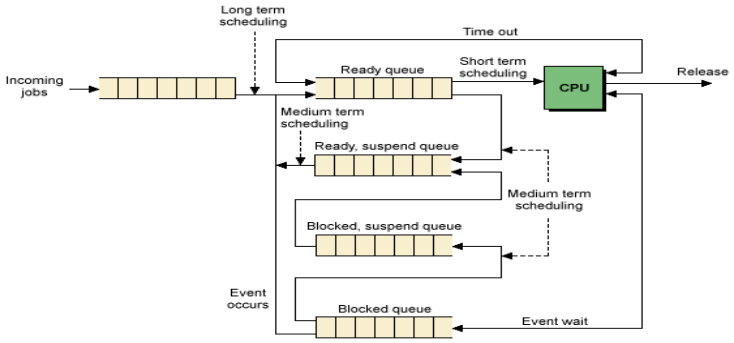
Keywords: CPU Scheduling, Round Robin Algorithm, Priority Algorithm, Turnaround Time, Waiting Time, Time Quantum, Context Switches.

المقدمة:

في علم الحاسوب، تعدّ الجدولة الوظيفة الأساسية لنظام التشغيل والتي تُحدد العملية (Process) التي سيتم تنفيذها، والعملية هي برنامج في حالة تنفيذ محمل على الذاكرة الرئيسة، وتعدّ كياناً فعالاً مع عداد البرنامج (Program counter PC) الذي يحمل عنوان الايعاز التالي المراد تنفيذه، وأن الغرض من جدولة العمليات هي اختيار عملية من العمليات الموجودة في الذاكرة الرئيسية وتخصيصها للمعالج لغرض التنفيذ، بحيث يكون المعالج دوماً مشغولاً. [1][7][15][18]

المعالج من أهم أجزاء الحاسوب والذي يقوم بتنفيذ العمليات ولكن يقوم بتنفيذ عملية واحدة في الوقت الواحد على الرغم من وجود الكثير من العمليات التي تحتاج إلى التنفيذ، إن المجدول (CPU Scheduler) هو المسؤول عن التنسيق بين العمليات باستخدام خوارزميات جدولة مختلفة، تحدث هذه الحالة عند وجود عمليتين أو أكثر في الوقت نفسه في حالة جاهزة (ready)، بحيث أن المجدول هو الذي يقوم باتخاذ القرار لاختيار واحدة من هذه العمليات لغرض التنفيذ من خلال منحها المعالج وتسمى الخوارزمية التي يستخدمها بخوارزمية الجدولة (Scheduling algorithm). [2][6][15]

نظم التشغيل قد تُعرِض بحدود ثلاثة أنواع متميزة من المُجدولِات، مجدول طويل الأمد، مجدول متوسط الأمد ومجدول قصير الأمد كما في الشكل (١). [4][15]



شكل (١)

أنواع المجدولات

- مجدول طويل الأمد (Long Term Scheduler) ومختصره (LTS) ويطلق عليه مجدول المهام (Job Scheduler) أيضاً ووظيفته: يختار العمليات جميعها من بركة المهام (Job Pool) وتحميلها إلى الذاكرة لغرض التنفيذ.
- مجدول قصير الأمد (Short Term Scheduler) ومختصره (STS) أو مجدول المعالج (CPU Scheduler) يختار عملية واحدة فقط جاهزة للتنفيذ من بين العمليات ويخصص لها المعالج.
- أما مجدول متوسط الأمد (Medium Term Scheduler) ومختصره (MTS) يزيل العمليات من الذاكرة وبذلك يحصل نقصان في درجة البرمجة المتعددة (Degree of Multiprogramming) بِعَدد العمليات المزالة من الذاكرة. [1][11][15][21][23]

الاعمال السابقة:

خوارزمية راوند روبن من الخوارزميات المستخدمة على نطاق واسع والتي تعتمد على شريحة زمنية ذات قيمة ثابتة، وإن هذه القيمة الثابتة تزيد من بعض معايير الجدولة وخاصة معدل الوقت الدوري، معدل زمن الانتظار، زمن الاستجابة وعدد حالات خزن السياقات، مما دفع الباحثين لايجاد خوارزميات مقترحة لمعالجة مساوئ هذه الخوارزمية.

H.S Behera وآخرون [13] استخدموا خوارزمية مقترحة هدفهم تقليل عدد حالات خزن السياقات الحاصلة في خوارزمية راوند روبن البسيطة من خلال استخدام اكثر من شريحة زمنية واحدة بالاعتماد على زمن التنفيذ وعلى عدد العمليات (فردي أم زوجي).

Samih M. Mostafa وآخرون أيضاً [24] استخدموا أسلوباً رياضياً من خلال المعادلات الرياضية.

واعتمد باحثون ومنهم Abbas Noon [4] أربع حالات دراسية بحيث كل حالة تتكون من أربع عمليات مع شريحة زمنية ثابتة وشريحة زمنية متغيرة وزمن وصول (صفر) وزمن وصول متفاوت.

AL.Hagery [16] اعتمد خوارزمية لغرض الاستغلال الأمثل للمعالج والحصول على كل من زمن انتظار وزمن استجابة ووقت دوري وعدد حالات خزن سياقات أقل.

Pallab Banerjee وباحثون آخرون [18]، سميت الخوارزمية المقترحة التي استخدموها بـ (AMRR) بحيث تم إستخدام شريحة زمنية متغيرة بدلاً من شريحة زمنية ثابتة لغرض معالجة مساوئ خوارزمية راوند روبن البسيطة.

أما الباحثان Ishwari Singh and Deepa Gupta فقد استخدما خوارزمية لتطوير أداء المعالج وتحسينه من خلال دمج خوارزميتي راوند روبن والأفضلية [15].

الباحث Siddharth Tyagi وآخرون [27]، تتلخص الخوارزمية المقترحة بخطوات تساعد في تقليل زمن الاستجابة، زمن الانتظار للعملية ذات الأفضلية الواطئة مما أدى إلى القضاء على حالة المجاعة (starvation).

Mohd Abdul Ahad [17] إعتمد خوارزمية مقترحة وبقيم افتراضية للتقليل من معدل الوقت الدوري، معدل زمن الانتظار وعدد حالات خزن السياقات.

باحث آخر اعتمد في حساب قيمة الشريحة الزمنية المتغيرة على أكبر زمن تنفيذ وأقل زمن تنفيذ للعملية من خلال جمع هاتين القيمتين وضرب حاصل جمعهما بـ (٠,٨) [8].

Saraj Hiranwal وباحث آخر [25] استخدما خوارزمية مقترحة لوجود مساوئ في خوارزمية راوند روبن البسيطة، بحيث تم استخدام أكثر من شريحة زمنية مما أدى إلى إلغاء مساوئ الخوارزمية.

معايير الجدولة Scheduling Criteria:

١- استغلال المعالج CPU Utilization

استغلال جميع وقت المعالج في تنفيذ العمليات، أي إن يكون المعالج مشغولاً بقدر الامكان ليتم استغلاله الاستغلال الأمثل (يكون عادة بين ٤٠٪-٩٠٪) [1][6][23][24]، عادة يمكن حسابه بالمعادلة الآتية: [19]

Processor utilization = (Processor busy time) / (Processor busy time + Processor idle time)

يمكن أن يُمثَّل بنسبة مئوية باستخدام إحدى المعادلتين:[3]

أ- نسبة استغلال المعالج = (وقت المعالج الكلي- الوقت الذي قضاه فارغاً) / (وقت المعالج الكلي) *١٠٠ .

ب- نسبة استغلال المعالج = (وقت التنفيذ للعمليات الكلي) / (وقت العمليات الكلي + الوقت المستغرق في تبديل المحتوى) *١٠٠ .

ومما يؤثر في هذا العامل حقيقة هو عدد مرات التبديل الذي يحصل، فكلما زاد هذا العدد كلما قلَّ استغلال المعالج وهذا منطقي جداً. ما نصبوا إليه هو أعلى استغلال ممكن للمعالج، أي عدم استغلال وقت فراغه فقط بل شغله بما ينفع أيضاً وعدم إضاعة وقت في عمليات التبديل. [2][3]

٢- الوقت الدوري Turnaround Time ومختصره (TAT):

الزمن المستغرق لتنفيذ عملية ما (من لحظة دخولها حتى لحظة انتهائها من التنفيذ) وهو مجموع الفترات الزمنية التي أمضاها في: [1][2][6][8][25]

أ- الانتظار قبل الدخول إلى الذاكرة.

ب- الانتظار في طابور الجاهز.

ج- التنفيذ على المعالج.

د- تنفيذ عمليات الإدخال والإخراج.

ويمكن حسابه من خلال المعادلة الآتية: [3][18][19]

TAT = Process completed time – Process arrival time

٣- الإنتاجية (العطاء) Throughput:

عدد العمليات التي يمكن إنجازها في فترة زمنية محددة [2][8][24][29].

ويمكن إيجادها من المعادلة الآتية: [1][19][23].

Throughput = (Number of processes completed) / (Time unit)

٤- زمن الانتظار Waiting Time ومختصره (WT):

الزمن الذي تستغرقه العملية في طابور الجاهز (Ready Queue) [8][24]، ويمكن إيجاده من المعادلة الآتية: [23].

Waiting Time = TAT – CPU burst time

٥- زمن الاستجابة Response Time ومختصره (RT):

الزمن المستغرق من دخول العملية إلى ظهور أول مخرج (استجابة) من العملية [24]، وهذا المعيار يعدّ ضرورياً للنظم التفاعلية [8][19][25]. ويمكن إيجاده من خلال المعادلة الآتية: [1][15][19][23].

RT = First response time – Time of submission of request

أهداف الجدولة Scheduling Objectives:

يسعى نظام التشغيل إلى تحسين أداء المعالج في أنظمة الزمن الحقيقي (Real Time) وأنظمة المشاركة الزمنية (Time Sharing) من خلال تحقيق الآتي: [7][11][15][20][27]

١. تقليل التبديل (خزن السياقات) Minimum Context Switches.
٢. زيادة استغلال المعالج Maximum CPU Utilization.
٣. زيادة الإنتاجية (العطاء) Maximum Throughput.
٤. تقليل الوقت الدوري Minimum Turnaround Time.
٥. تقليل زمن الانتظار Minimum Waiting Time.
٦. تقليل زمن الاستجابة Minimum Response Time.
٧. زيادة كفاءة المجدول Maximum Scheduler Efficiency.
٨. منع حصول المجاعة (الموت جوعاً) Avoid Starvation.

أنواع الجدولة Scheduling Types:

يوجد نوعان من الجدولة هما:

١- جدولة بتدخل (Preemptive Scheduling):

نظام التشغيل يتدخل في سحب المعالج من العملية المنفذة حاليا لسبب ما، كأن يكون وصول عملية جديدة بأفضلية أعلى أو لسبب آخر قبل أن تنتهي العملية من التنفيذ، وبذلك يقوم نظام التشغيل بمنح المعالج للعملية الجديدة لغرض البدء بالتنفيذ أيضاً. [4][6][9][26][28]

٢- جدولة من دون تدخل (Non Preemptive Scheduling):

نظام التشغيل لا يقوم بسحب المعالج من العملية المنفذة حالياً إلى أن ينتهي تنفيذها على الرغم من وصول عملية جديدة بأفضلية أعلى مثلاً، بحيث إنّ العملية تعيد المعالج بشكل طوعي إلى النظام. [4][6][9][26][28]

خوارزميات الجدولة Scheduling Algorithms:

هنالك مجموعة من الخوارزميات منها:

١- خوارزمية القادم أولاً يُخدم أولاً (FCFS) .First Come First Serve Alg

جدولة هذه الخوارزمية من النوع (جدولة من دون تدخل)، بحيث يتم منح المعالج للعملية التي تصل طابور الجاهز أولاً. المعالج يُنتزع من العملية عند الانتهاء من التنفيذ نهائيّاً أو بتعبير آخر عندما ينتهي الوقت المخصص للتنفيذ (burst time) بشكل تام. مميزاتها عادلة وبطيئة. [5][10][11][14][17][22]

٢- خوارزمية الزمن الأقصر أولاً Shortest Job First Alg. (SJF)

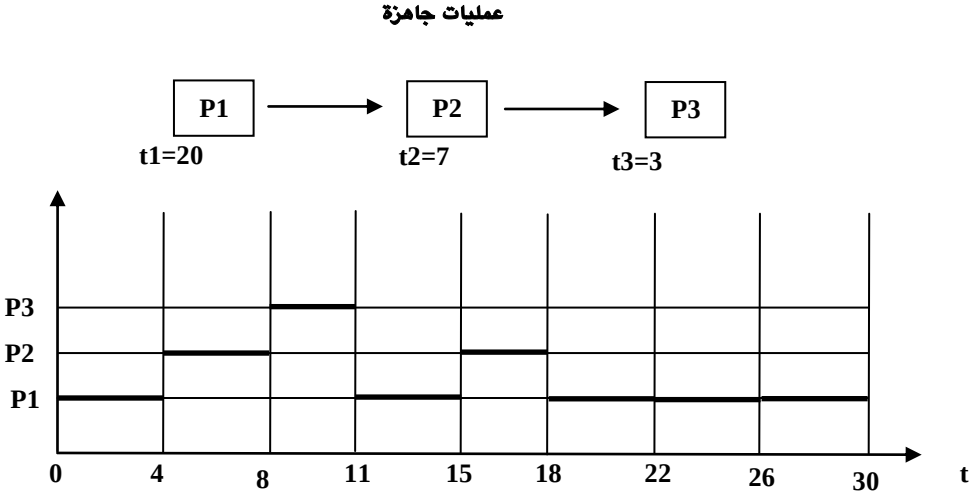
جدولة هذه الخوارزمية من النوع (جدولة من دون تدخل) أيضاً، بحيث أنّ العملية التي تمتلك زمن تنفيذ أقصر تحصل على المعالج أولاً. لا يتم سحب المعالج من العملية الحالية إلى أن ينتهي الزمن المخصص لها بالكامل. في حالة تساوي أكثر من عملية في زمن التنفيذ يتم التعامل معهم بخوارزمية القادم أولاً يُخدم أولاً. [5][17][22][23]

٣- خوارزمية راوند روبن Round Robin Alg. (RR)

صُممت هذه الخوارزمية خصيصاً لأنظمة المشاركة الزمنية وجدولتها من النوع (جدولة بتدخل)، وتعدّ خوارزمية محسَّنة لخوارزمية القادم أولاً يُخدم أولاً مدعومة بشريحة زمنية معينة وثابتة (Time Quantum) تتراوح عموماً بين (10-100 ms)، تمتلك هذه الخوارزمية طابور من النوع الدائري بحيث أنَّ العملية التي يُقطع تنفيذها بالاعتماد على قيمة الشريحة الزمنية يتم وضعها في نهاية الطابور الدائري إذا كان زمن التنفيذ أكبر من قيمة الشريحة الزمنية وإلاّ تخرج العملية من النظام إذا كان زمن التنفيذ أقل من قيمة الشريحة الزمنية أو في حالة تساوي القيمتين. [4][8][11][13] [14][16][17][19][23]

أداء الخوارزمية يعتمد بشدة على قيمة الشريحة الزمنية، الشريحة الزمنية الكبيرة تزيد من زمن الاستجابة وتسلك الخوارزمية سلوك خوارزمية القادم أولاً يُخدم أولاً، أما الشريحة الزمنية القصيرة جداً تُزيد من حالات التبديل (خزن السياقات)

مما يشكل عبئاً إضافياً مع زمن استجابة قصير، الشكل (٢) أدناه يوضح مثال للخوارزمية وبشريحة زمنية مقدارها (4 ms). [12]



شكل (٢)

مثال لخوارزمية راوند روبن

٤- خوارزمية الأفضلية Priority Algorithm:

الخوارزمية لها نوعان من الجدولة (جدولة بتدخل وجدولة من دون تدخل)، كل عملية تمتلك رقماً يمثل أفضليتها بحيث أن الرقم الصغير يمثل الأفضلية الأعلى والعكس صحيح أيضاً. في حالة تساوي أكثر من عملية بالأفضلية يتم التعامل معهم بخوارزمية القادم أولاً يُخدم أولاً. [5][6][10][11][16][22][23]

الخوارزمية المقترحة Proposed Algorithm:

معمارية الخوارزمية المقترحة تتركز بمساويء خوارزمية راوند روبن البسيطة التي تعطي أفضلية متساوية لكل العمليات (القادم أولاً يُخدم أولاً) من خلال شريحة زمنية معينة ثابتة (static) خلال مدة التنفيذ، بسبب مساويء الخوارزمية البسيطة

وخاصة مع العمليات ذات زمن تنفيذ قصير جداً التي تؤدي إلى زيادة في زمن الانتظار وزمن الاستجابة للعمليات وبالتالي تقليل من إنتاجية (عطاء) النظام.

الخوارزمية المقترحة تمر بمجموعة إجراءات أو خطوات لإيجاد أكثر من شريحة زمنية متغيرة (dynamic) لغرض اعتمادها في التنفيذ لوضع علاج لمساويء خوارزمية راوند روبن البسيطة بحيث أنَّ الشريحة الزمنية المتغيرة تساعد في تقليل زمن الاستجابة، تقليل زمن الانتظار للعمليات ذات الأفضلية الواطئة مع إعطاء فرصة عادلة لتلك العمليات لغرض الحصول على مصادر الحاسبة، فضلاً عن وقت دوري قليل مع خزن سياقات أقل مقارنة بالخوارزمية البسيطة.

يمكن أن نلخّص اجراءات ايجاد الشريحة الزمنية المتغيرة بمجموعة خطوات:

١- ترتيب العمليات بشكل تصاعدي في طابور الجاهز بالاعتماد على رقم الافضلية.

٢- إيجاد المعدل لزمن التنفيذ للعمليات.

٣- إيجاد **قيمة الشريحة الزمنية المتغيرة** بالاعتماد على عدد العمليات (n) كما يأتي:

٣-١: إذا كانت **عدد العمليات عدداً فردياً** فانّ قيمة الشريحة الزمنية تساوي حاصل **جمع المعدل لزمن التنفيذ وأكبر زمن تنفيذ للعمليات مقسوماً على (٢)**.

٣-٢: إذا كانت **عدد العمليات عدداً زوجياً** فإنّ قيمة الشريحة الزمنية تساوي حاصل **جمع زمن التنفيذ للعملية في الترتيب (n/2) والعملية في الترتيب ((n/2)+1) مقسوماً على (٢) والناتج يجمع مع المعدل لزمن التنفيذ للعمليات مقسوماً على (٢) أيضاً**.

٤- تنفيذ جميع العمليات ولمرة واحدة بالشريحة الزمنية المتغيرة المحسوبة أعلاه.

٥- ترتيب العمليات بشكل تصاعدي بالاعتماد على المتبقي من زمن التنفيذ.

٦- تكرار الخطوات (٢، ٣، ٤، ٥) لما تبقّى من العمليات ولما تبقّى من زمن التنفيذ إلى أن تصل قيمة زمن تنفيذ العمليات إلى (صفر).

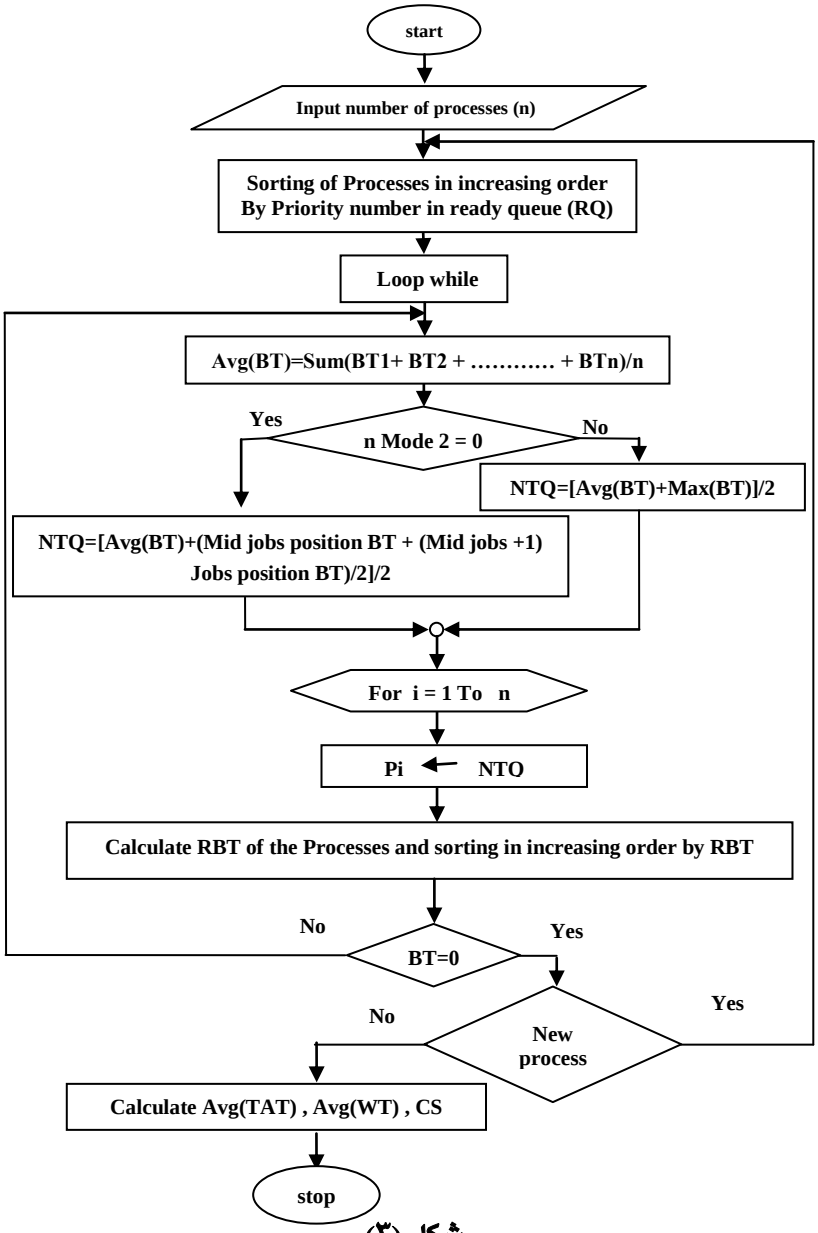
المصطلحات المستخدمة في الخوارزمية المقترحة:

يبين الجدول الآتي مجموعة من المصطلحات المستخدمة في الخوارزمية المقترحة:

جدول (١)

المصطلحات المستخدمة في الخوارزمية المقترحة

ت	اسم المصطلح بالعربي	اسم المصطلح بالإنكليزي	المختصر بالإنكليزي
١	طابور الجاهز	Ready Queue	RQ
٢	الشريحة الزمنية	Time Quantum	TQ
٣	الشريحة الزمنية الجديدة	New Time Quantum	NTQ
٤	زمن تنفيذ العملية	CPU Burst Time	BT
٥	عدد العمليات	Number of Processes	N
٦	خزن السياقات (التبديل)	Context Switches	CS
٧	زمن الوصول للعملية	Arrival Time	AT
٨	زمن انتهاء تنفيذ العملية	Completed Time	CT
٩	معدل زمن تنفيذ العمليات	Burst Time Average	Avg(BT)
١٠	المتبقي من زمن التنفيذ	Remaining Burst Time	RBT
١١	الوقت الدوري	Turnaround Time	TAT
١٢	معدل الوقت الدوري	Average Turnaround Time	Avg(TAT)
١٣	زمن الانتظار	Waiting Time	WT
١٤	معدل زمن الانتظار	Average Waiting Time	Avg(WT)



شكل (٣)

المخطط الانسيابي للخوارزمية المقترحة

حالات دراسية:

مثال ١:

على افتراض مجموعة من العمليات وبشريحة زمنية قيمتها (25 ms) وكما في الجدول (٢) وان القيم مأخوذة من experiment 1 [27]:

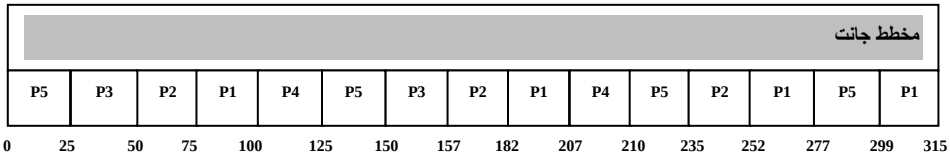
جدول (٢)

مجموعة من العمليات (Snapshot)

Processes	Priority no.	CPU Burst Time
P1	6	91
P2	4	67
P3	3	32
P4	7	28
P5	1	97

الحل:

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة (25 ms) **تم الحصول على مخطط جانت كما في الشكل (٤)**

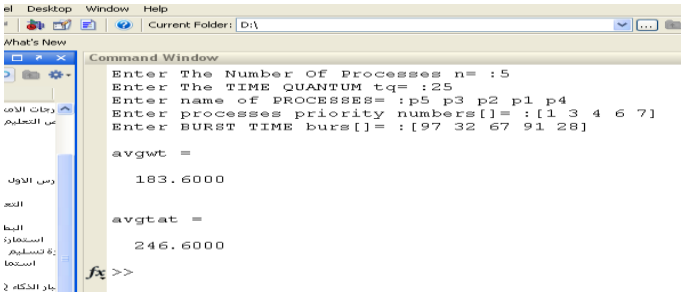


شكل (٤)

مخطط جانت للمثال (١) باستخدام الخوارزمية البسيطة

تم الحصول على النتائج العملية أدناه باستخدام برنامج ماتلاب وكما في الشكل (٥)

Avg(TAT) = 246.6 ms Avg(WT) = 183.6 ms CS = 14



شكل (٥)

النتائج العملية للمثال (١) باستخدام الخوارزمية البسيطة

الحل:

على وفق الخوارزمية المقترحة:

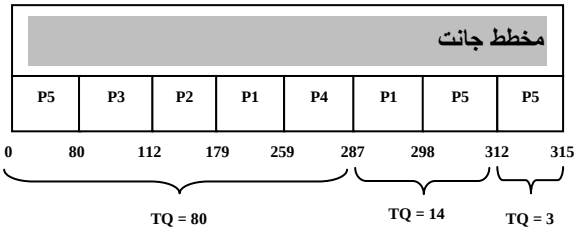
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (٣)

جدول (٣)

مجموعة العمليات للمثال (١) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	CPU Burst Time
P5	1	97
P3	3	32
P2	4	67
P1	6	91
P4	7	28

من الجدول اعلاه تم الحصول على مخطط جانت وكما في الشكل (٦)

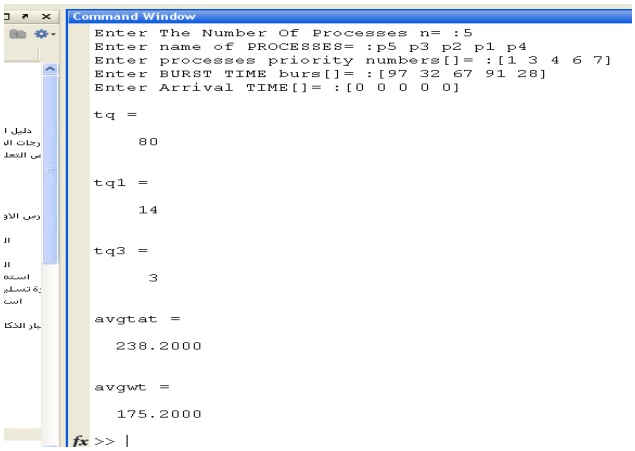


شكل (٦)

مخطط جانت للمثال (١) باستخدام الخوارزمية المقترحة

وتم الحصول على النتائج العملية أدناه باستخدام برنامج ماتلاب وكما في الشكل (٧)

Avg(TAT) = 238.2 ms Avg(WT) = 175.2 ms CS = 6



شكل (٧)

النتائج العملية للمثال (١) باستخدام الخوارزمية المقترحة

جدول (٤)

الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (١)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	25	246.6	183.6	14
الخوارزمية المقترحة	80, 14, 3	238.2	175.2	6
مقدار التحسين بين الخوارزميتين		8.4	8.4	8

مثال ٢:

على افتراض مجموعة من العمليات وبشريحة زمنية قيمتها (25 ms) وكما في الجدول (٥) وان القيم مأخوذة من case 1 [13]:

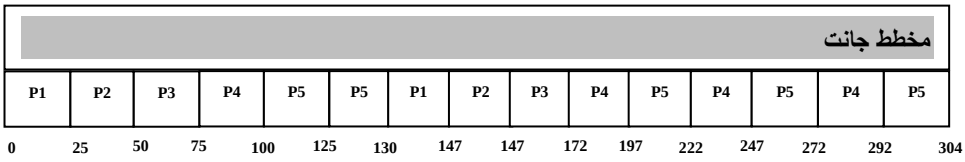
جدول (٥)

مجموعة من العمليات (snapshot)

Processes	Priority no.	CPU Burst Time
P1	1	30
P2	2	42
P3	3	50
P4	4	85
P5	5	97

الحل:

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة 25 (ms)، **تم الحصول على مخطط جانت كما في الشكل (٨)**



شكل (٨)

مخطط جانت للمثال (٢) باستخدام الخوارزمية البسيطة

Avg(TAT) = 207 ms Avg(WT) = 146.2 ms CS = 13

الحل:

على وفق الخوارزمية المقترحة:

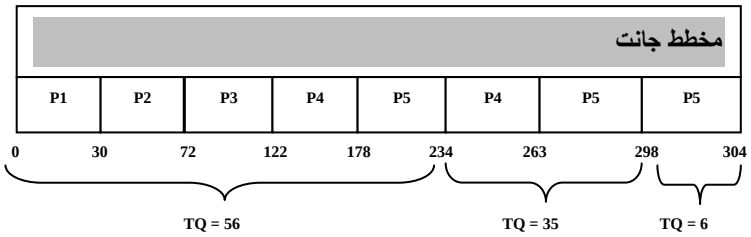
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (٦)

جدول (٦)

مجموعة العمليات للمثال (٢) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	CPU Burst Time
P3	1	30
P1	2	42
P4	3	50
P2	4	85
P5	5	97

من الجدول (٦) تم الحصول على مخطط جانت وكما في الشكل (٩)



شكل (٩)

مخطط جانت للمثال (٢) باستخدام الخوارزمية المقترحة

Avg(TAT) = 158.2 ms Avg(WT) = 97.4 ms CS = 6

جدول (٧)

الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (٢)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	25	207	146.2	13
الخوارزمية المقترحة	56,35,6	158.2	97.4	6
مقدار التحسين بين الخوارزميتين		48.8	48.8	7

مثال ٣ :

على افتراض مجموعة من العمليات وبشريحة زمنية قيمتها (5 ms) وكما في الجدول (٨) وأن القيم مأخوذة من case 1 [15]:

جدول (٨)

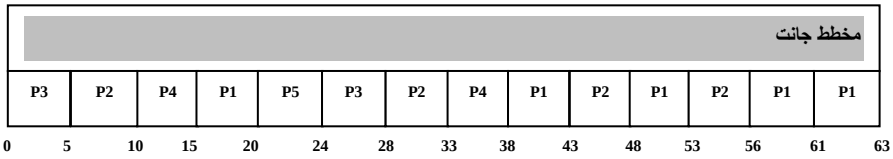
مجموعة من العمليات (snapshot)

Processes	Priority no.	CPU Burst Time
P1	4	22
P2	2	18
P3	1	9
P4	3	10
P5	5	4

الحل:

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة (5 ms)

تم الحصول على مخطط جانت كما في الشكل (١٠)



شكل (١٠)

مخطط جانت للمثال (٣) باستخدام الخوارزمية البسيطة

Avg(TAT) = 41.8 ms Avg(WT) = 29.2 ms CS = 12

الحل:

على وفق الخوارزمية المقترحة:

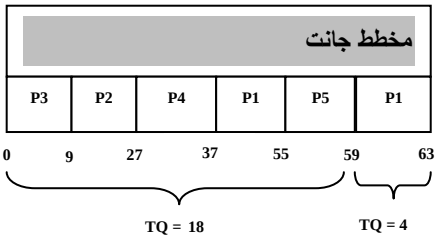
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (٩)

جدول (٩)

مجموعة العمليات للمثال (٣) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	CPU Burst Time
P3	1	9
P2	2	18
P4	3	10
P1	4	22
P5	5	4

من الجدول (٩) تم الحصول على مخطط جانت وكما في الشكل (١١)



شكل (١١)

مخطط جانت للمثال (٣) باستخدام الخوارزمية المقترحة

Avg(TAT) = 39 ms Avg(WT) = 26.4 ms CS = 5

جدول (١٠)

الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (٣)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	5	41.8	29.2	12
الخوارزمية المقترحة	18,4	39	26.4	5
مقدار التحسين بين الخوارزميتين		2.8	2.8	7

مثال ٤:

على افتراض مجموعة من العمليات وبشريحة زمنية قيمتها (20 ms) وكما في الجدول (١١) وان القيم مأخوذة من case 1 [18]:

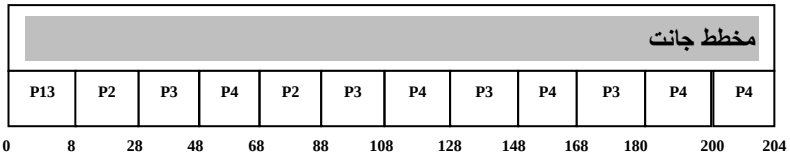
جدول (١١)

مجموعة من العمليات (snapshot)

Processes	Priority no.	CPU Burst Time
P1	1	8
P2	2	40
P3	3	72
P4	4	84

الحل :

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة (20 ms) **تم الحصول على مخطط جانت كما في الشكل (١٢)**



شكل (١٢)

مخطط جانت للمثال (٤) باستخدام الخوارزمية البسيطة

Avg(TAT) = 120 ms Avg(WT) = 69 ms CS = 10

الحل:

على وفق الخوارزمية المقترحة

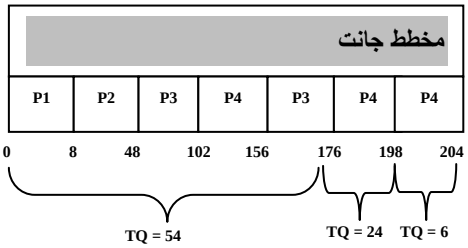
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (١٢)

جدول (١٢)

مجموعة العمليات للمثال (٤) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	CPU Burst Time
P1	1	8
P2	2	40
P3	3	72
P4	4	84

من الجدول السابق تم الحصول على مخطط جانت وكما في الشكل (١٣)



شكل (١٣)

مخطط جانت للمثال (٤) باستخدام الخوارزمية المقترحة

Avg(TAT) = 108.5 ms Avg(WT) = 57.5 ms CS = 5

جدول (١٣)

الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (٤)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	20	120	69	10
الخوارزمية المقترحة	54,24,6	108.5	57.5	5
مقدار التحسين بين الخوارزميتين		11.5	11.5	5

مثال ٥:

على افتراض مجموعة من العمليات وبشريحة زمنية قيمتها (20 ms) وكما في الجدول (١٤):

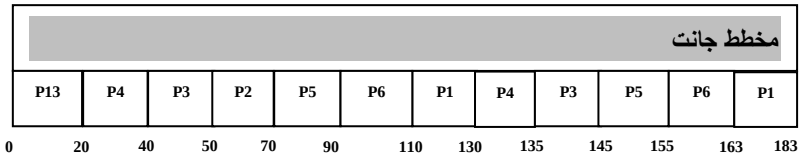
جدول (١٤)

مجموعة من العمليات (snapshot)

Processes	Priority no.	CPU Burst Time
P1	1	60
P2	4	10
P3	3	30
P4	2	25
P5	7	30
P6	8	28

الحل:

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة (20 ms) **تم الحصول على مخطط جانت كما في الشكل (١٤)**

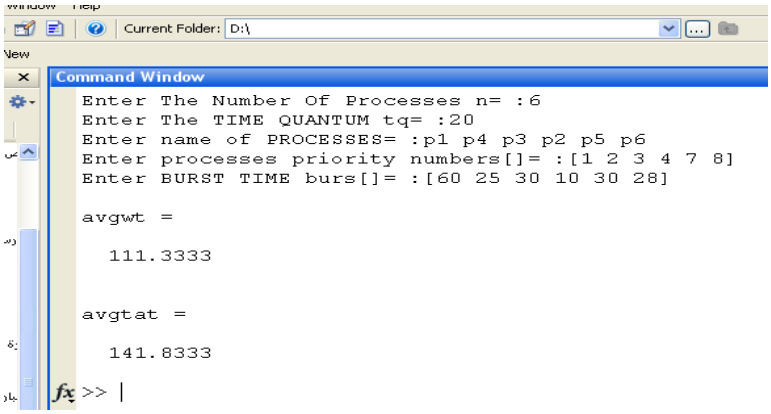


شكل (١٤)

مخطط جانت للمثال (٥) باستخدام الخوارزمية البسيطة

وتم الحصول على النتائج العملية أدناه باستخدام برنامج ماتلاب وكما في الشكل (١٥)

Avg(TAT) = 141.8 ms Avg(WT) = 111.3 ms CS = 11



شكل (١٥)

النتائج العملية للمثال (١٥) باستخدام الخوارزمية البسيطة

الحل:

على وفق الخوارزمية المقترحة

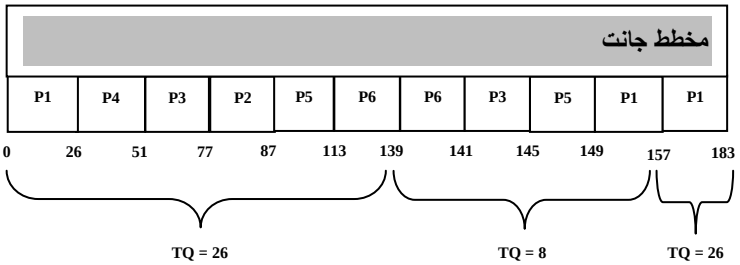
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (١٥)

جدول (١٥)

مجموعة العمليات للمثال (٥) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	CPU Burst Time
P1	1	60
P4	2	25
P3	3	30
P2	4	10
P5	7	30
P6	8	28

من الجدول (١٥) تم الحصول على مخطط جانت وكما في الشكل (١٦)

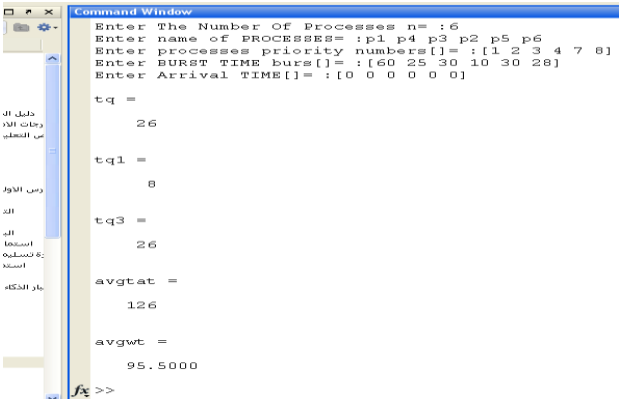


شكل (١٦)

مخطط جانت للمثال (٥) باستخدام الخوارزمية المقترحة

وتم الحصول على النتائج العملية أدناه باستخدام برنامج ماتلاب وكما في الشكل (١٧)

Avg(TAT) = 126 ms Avg(WT) = 95.5 ms CS = 8



شكل (١٧)

النتائج العملية للمثال (٥) باستخدام الخوارزمية المقترحة

جدول (١٦)

الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (٥)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	20	141.8	111.3	11
الخوارزمية المقترحة	26,8,26	126	95.5	8
مقدار التحسين بين الخوارزميتين		15.8	15.8	3

مثال ٦ :

على افتراض مجموعة من العمليات وبشريحة زمنية قيمتها (10 ms) وكما في الجدول (١٧) وان القيم مأخوذة من 4 example [17]:

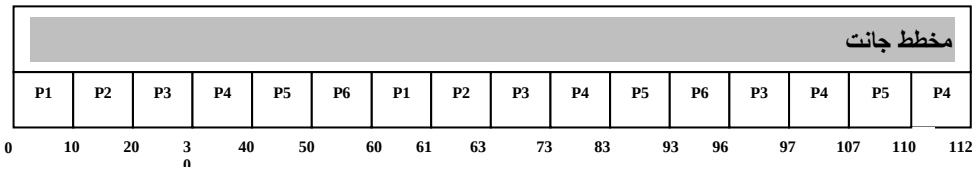
جدول (١٧)

مجموعة من العمليات (snapshot)

Processes	Priority no.	CPU Burst Time
P1	1	11
P2	2	12
P3	3	21
P4	4	32
P5	5	23
P6	6	13

الحل :

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة (10 ms) **تم الحصول على مخطط جانت كما في الشكل** (١٨)



شكل (١٨)

مخطط جانت للمثال (٦) باستخدام الخوارزمية البسيطة

Avg(TAT) = 89.83 ms Avg(WT) = 71.16 ms CS = 15

الحل:

على وفق الخوارزمية المقترحة

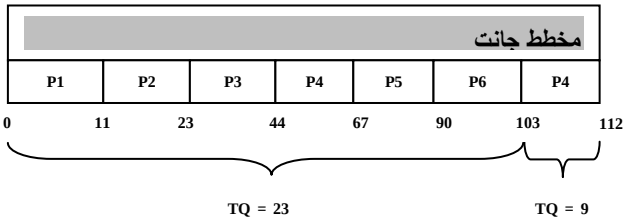
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (١٨)

جدول (١٨)

مجموعة العمليات للمثال (٦) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	CPU Burst Time
P1	1	11
P2	2	12
P3	3	21
P4	4	32
P5	5	23
P6	6	13

من الجدول (١٨) تم الحصول على مخطط جانت وكما في الشكل (١٩)



شكل (١٩)

مخطط جانت للمثال (٦) باستخدام الخوارزمية المقترحة

Avg(TAT) = 63.83 ms Avg(WT) = 45.16 ms CS = 6

جدول (١٩)

الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (٦)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	10	89.83	71.16	15
الخوارزمية المقترحة	23,9	63.83	45.16	6
مقدار التحسين بين الخوارزميتين		26	26	9

مثال ٧:

على افتراض مجموعة من العمليات وبشريحة زمنية قيمته (15 ms) وكما في الجدول (٢٠) وأن القيم افتراضية:

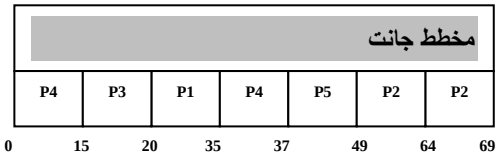
جدول (٢٠)

مجموعة من العمليات (snapshot)

Processes	Priority no.	Arrival Time	CPU Burst Time
P1	2	7	15
P2	5	0	20
P3	1	10	5
P4	3	0	17
P5	4	3	12

الحل:

على وفق الخوارزمية البسيطة (راوند روبن والأفضلية) مع شريحة زمنية ثابتة (15ms)
تم الحصول على مخطط جانت كما في الشكل (٢٠)



شكل (٢٠)

مخطط جانت للمثال (٧) باستخدام الخوارزمية البسيطة

Avg(TAT) = 38 ms Avg(WT) = 24.2 ms CS = 5

الحل:

على وفق الخوارزمية المقترحة

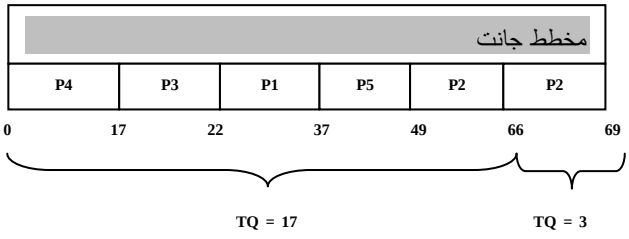
في البداية يتم ترتيب العمليات تصاعدياً على وفق الأفضلية وكما في الجدول (٢١)

جدول (٢١)

مجموعة العمليات للمثال (٧) مرتبة تصاعدياً على وفق الأفضلية

Processes	Priority no.	Arrival Time	CPU Burst Time
P3	1	10	5
P1	2	7	15
P4	3	0	17
P5	4	3	12
P2	5	0	20

من الجدول (21) تم الحصول على مخطط جانت وكما في الشكل (21)

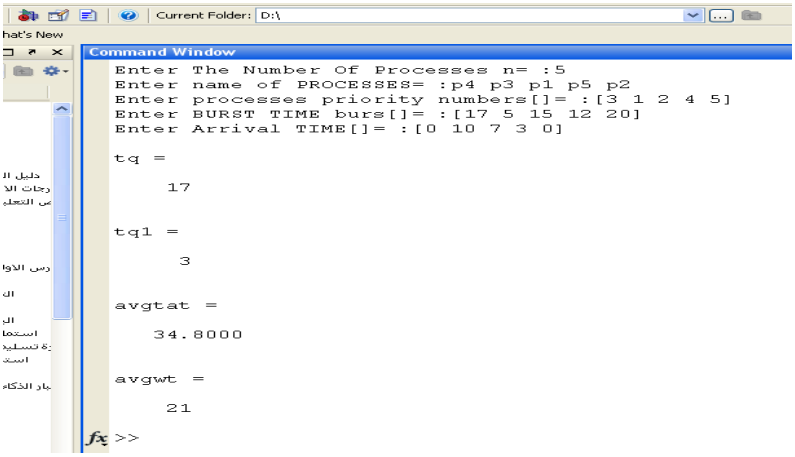


شكل (٢١)

مخطط جانت للمثال (٧) باستخدام الخوارزمية المقترحة

وتم الحصول على النتائج العملية أدناه باستخدام برنامج ماتلاب وكما في الشكل (٢٢)

Avg(TAT) = 34.8 ms Avg(WT) = 21 ms CS = 4



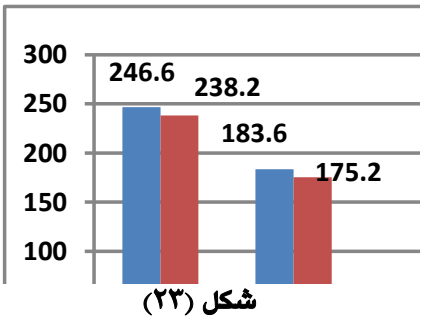
شكل (٢٢)

النتائج العملية للمثال (٧) باستخدام الخوارزمية المقترحة

جدول (٢٢)

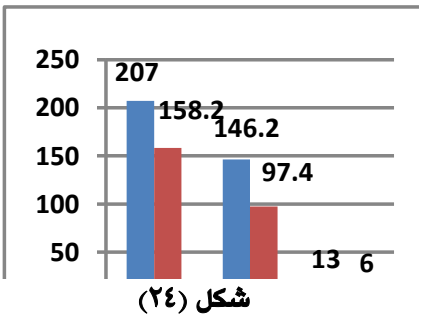
الفرق بين نتائج الخوارزمية البسيطة والخوارزمية المقترحة للمثال (٧)

نوع الخوارزمية	الشريحة الزمنية	معدل الوقت الدوري	معدل زمن الانتظار	عدد خزن السياقات
الخوارزمية البسيطة	15	38	24.2	5
الخوارزمية المقترحة	17, 3	34.8	21	4
مقدار التحسين بين الخوارزميتين		3.2	3.2	1



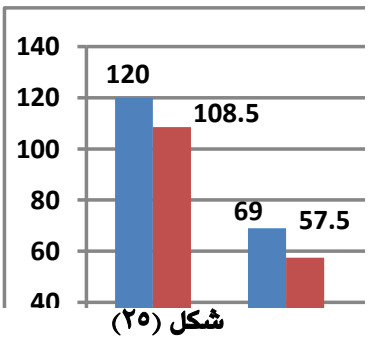
شكل (٢٣)

الفرق في معايير الجدولة بين الخوارزميتين للمثال ١



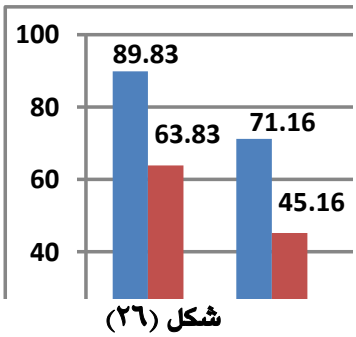
شكل (٢٤)

الفرق في معايير الجدولة بين الخوارزميتين للمثال ٢



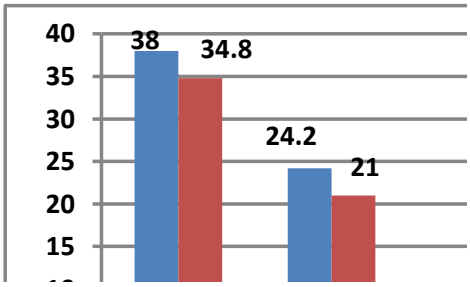
شكل (٢٥)

الفرق في معايير الجدولة بين الخوارزميتين للمثال ٤



شكل (٢٦)

الفرق في معايير الجدولة بين الخوارزميتين للمثال ٦



شكل (٢٧)

الفرق في معايير الجدولة بين

الخوارزميتين للمثال ٧

الاستنتاجات:

١- **الخوارزمية المقترحة أفضل أداءً من الخوارزمية البسيطة من خلال النتائج التي تشير إلى انخفاض في معياري معدل الوقت الدوري ومعدل زمن الانتظار فضلاً عن انخفاض في معيار عدد حالات خزن السياقات الموضَّحة في الجداول ٤، ٧، ١٠، ١٣، ١٦، ١٩، ٢٢، وكذلك من خلال الأشكال ٢٣، ٢٤، ٢٥، ٢٦، ٢٧، لخمسة أمثلة من أصل سبعة أمثلة.**

٢- **نلاحظ أنّ مقدار التحسين بين الخوارزميتين في معياري معدل الوقت الدوري ومعدل زمن الانتظار هو رقم ثابت للمثال نفسه وكما موضح في الجداول ٤، ٧، ١٠، ١٣، ١٦، ١٩، ٢٢.**

٣- **القضاء على حالة المجاعة (starvation) للعمليات التي ذات أفضلية واطئة من خلال تقنية التقادم (aging) وذلك بإعطاء أفضلية أعلى للتنفيذ بالاعتماد على المتبقي من زمن التنفيذ (RBT).**

٤- **تنفيذ عادل لجميع العمليات من خلال مفهوم الشريحة الزمنية المتغيرة (dynamic) مع وجود الأفضلية.**

٥- المثال (١) في الخوارزمية المقترحة في هذا البحث تم الحصول على معدل الوقت الدوري (238.2) بينما في [27] كان (256.5) وبفارق (18.3 ms)، معدل زمن الانتظار(175.2) بينما في [27] كان (179.7) وبفارق مقداره (4.5 ms)، عدد خزن السياقات (6) بينما في [27] كان العدد(7) وبفارق مقداره (1).

٦- المثال (٢) في الخوارزمية المقترحة في هذا البحث تم الحصول على معدل الوقت الدوري (158.2) بينما في [13] كان (195.2) وبفارق (37 ms)، معدل زمن الانتظار(97.4) بينما في [13] كان (134.4) وبفارق مقداره (37 ms)، عدد خزن السياقات (6) بينما في [13] كان العدد(7) وبفارق مقداره (1).

٧- المثال (٣) في الخوارزمية المقترحة في هذا البحث تم الحصول على معدل الوقت الدوري (39) بينما في [15] كان (38.8) وهي قيم متقاربة، معدل زمن الانتظار(26.4) بينما في [15] كان (26.2) وهي قيم متقاربة أيضاً، عدد خزن السياقات (5) بينما في [15] كان العدد(8) وبفارق مقداره (3).

٨- المثال (٤) في الخوارزمية المقترحة في هذا البحث تم الحصول على معدل الوقت الدوري (108.5) بينما في [18] كان (112) وبفارق (3.5 ms)، معدل زمن الانتظار(57.5) بينما في [18] كان (61) وبفارق مقداره (3.5 ms)، عدد خزن السياقات (5) بينما في [18] كان العدد(5) وبفارق مقداره (صفر).

٩- المثال (٦) في الخوارزمية المقترحة في هذا البحث تم الحصول على معدل الوقت الدوري (63.83) بينما في [17] كان (71) وبفارق (7.17 ms)، معدل زمن الانتظار(45.16) بينما في [17] كان (52.33) وبفارق مقداره (7.17 ms)، عدد خزن السياقات (6) بينما في [1] كان العدد(11) وبفارق مقداره (5).

التوصيات :

١- تطبيق الخوارزمية مع عدد كبير من العمليات ربّما تعطي نتائج أفضل.

٢- إمكانية تطوير الخوارزمية المقترحة نفسها مع العمليات ذات زمن تنفيذ قصير عند ورودها خلف عمليات ذات زمن تنفيذ طويل وبالاعتماد على زمن التنفيذ بالترتيب التصاعدي مع عدم وجود رقم أفضلية تنفيذ للعمليات.

٣- إمكانية تدريس الخوارزمية في المناهج الدراسية لمادة نظم التشغيل.

قائمة المصادر

المصادر باللغة العربية

[١] الأفندي، عبدالرحمن حمد (٢٠٠٩)، **مفاهيم نظم التشغيل**، كلية الدراسات التطبيقية وخدمة المجتمع، جامعة الملك فيصل.

[٢] اللبني، المهندس يمان والمهندس أسامة العبد الله (٢٠٠٥)، **تصميم وتنفيذ نظم التشغيل الحديثة**، ط ١، سورية، شعاع للنشر والعلوم ، ص ١٥١-١٧١.

[٣] White ، Laith white (١٤٢٩/١٤٢٨) هـ ، **نظم التشغيل الالكتروني**، جامعة الملك سعود، قسم تقنية المعلومات، ص ٩٠.

المصادر باللغة الاجنبية

[4] Abbas Noon, Ali Kalakech and Seifedine Kadry (May 2011), A New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average, (IJCSI) International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, ISSN(Online):1694-0814, PP. 224-229, Lebanon. www.IJCSI.org.

[5] Abhishek Chaudhary, Saoud Sarwar and Rohit Singh Yadav (Octobar 2012), optimization of the performance of cpu scheduling, vsrd International Journal of Computer Science and Information Technology ,Vol. 2, No. 10, PP. 854-856, India.

[6] Abraham Siberschatz, Peter Baer Galvin and Greg Gagne (Copyright 2013), Operating System Concepts, 9th edition, John Wiley and Sons.

[7] Ajit Singh, Priyanka Goyal and Sahil Batra (2010), An Optimized Round Robin Scheduling Algorithm for CPU Scheduling, (IJCSE) International Journal On Computer Science and Engineering,Vol. 02, No. 07, 2383-2385, ISSN:0975-3397.

[8] Ali Jbaeer Dawood (2012), Improving Efficiency of Round Robin Scheduling Using Ascending Quantum And Minumim – Maxumum Burst Time, J. of University of anbar for pure science: Vol. 6:No. 2, ISSN:1991- 8941, Iraq.

[9] Amit Kumar Sain (2013), Dynamical Modified R.R. CPU Scheduling Algorithm, International Journal of Computer Trends and Technology, V. 4, Issue 2, PP. 90-93, ISSN:2231-2803. http://www.internationaljournalssrg.org

[10] Ankur Bhardwaj, Rachhpal singh and Gaurav (April-May 2013), Comparative Study of Scheduling Algorithms in Operating System, International Journal of Computers and Distributed Systems, Vol. No.3, Issue 1, ISSN:2278-5183, PP. 5-7. www.ijedsonline.com.

[11] Dinesh Gupta, Sakshi Gupta and Neha Jhamb (2012), Minimizing Response Time And Effective Utilization of I/O-bound Processes using Approximate Zero Response Algorithm", Int.J. Computer Technology Applications, Vol. 3(2), PP.611-616, ISSN: 2229-6093.

[12] Francis Cottet, Joelle Delacroix and Zoubir Mammeri (Copyright 2002), Scheduling In Real Time Systems, France.

[13] H.S. Behera, R. Mahanty and Debashree Nayak (August 2010), A New Proposed Dynamic Quantum With Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis, International Journal of Computer Applications (0975-8887), V. 5, No. 5, PP. 10-15, India.

[14] H.S. Behera, Sreelipa Curtis and bijayalaxmi panda (March 2012), enhancing the cpu performance using a modified mean-deviation round robin scheduling algorithm for real time systems, (JGRCS) Journal of Global Research in Computer Science ,V. 3, No. 3, ISSN: 2229-371X, India.

[15] Ishwari Singh and Deepa Gupta (Oct 2012), A Priority based Round Robin CPU Scheduling Algorithm for Real Time Systems, (IJIET) International Journal of Innovations in Engineering and Technology,V. 1, Issue 3, ISSN: 2319-1058, India.

[16] Mohammed Abdullah Hassan Al.Hagery (2011), A selective Quantum of Time For Round Robin Algorithm to Increase CPU Utilization, International Journal of Computer Information Systems ,Vol. 3, No. 2, ISSN: 2229-5208, PP. 54-59, Qassim, Kingdom of Saudi Arabia.

[17] Mohd Abdul Ahad (November 2012), Modifying Round Robin Algorithm for Process Scheduling using Dynamic Quantum Precision, Special Issue of International Journal of Computer Applications (0975-8887) on Issues and Challenges in Networking Intelligence and Computing Technologies-ICNICT 2012, India.

[18] Pallab Banerjee, Probal Banerjee and Shweta Sonali Dhal (August 2012), Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using static Time Quantum, (IJITEE) International Journal of Innovative Technology and Exploring Engineering, V. 1, Issue 3, ISSN:2278-3075.

[19] P. Surendra Varma (November 2012), A Best Possible Time quantum for Improving Shortest Remaining Burst Round Robin (SRBRR) Algorithm, International Journal of Advanced Research in Computer Science and Sotfware Engineering, V. 2, Issue 11, ISSN:2277-128X, PP. 228-237, India.

[20] Rakesh Kumar, Abhishek K. Mishra, Navin Prakash and Himanshu Sharma (2010), An Improved Round Robin Scheduling Algorithm for CPU Scheduling, (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 04, 1064-1066, ISSN:0975-3397, India.

[21] Rakesh Kumar Yadav and Anurag Upadhayay (July 2012), A fresh loom for Multilevel Feedback Queue Scheduling Algorithm, International Journal of Advances in Engineering Science,Vol. 2, Issue 3, PP. 21-23, Print-ISSN:2231-2013 e-ISSN: 2231-0347, India.

[22] Renu Tiwari and Dr. Naveen Hemrajani (February 2013), REN an Efficient Algorithm for Process Scheduling to Optimize CPU Blocking With the Time Slice, (IJCSITS) IRACST – International Journal of Computer Science and Information Technology and Security,Vol. 3, No. 1, PP. 79- 83, ISSN:2249-9555, India.

[23] Ryan Richard Guadans, Maria Rona Perez and Larry Rutaquio Ir (January 2013), A Comprehensive Review for Central Processing Unit Scheduling Algorithm, (IJCSI) International Journal of Computer Science Issue,Vol. 10, Issue 1, No. 2, ISSN (Print): 1694-0784, Issue (Online): 1694-0814, PP. 353-358. www.IJCSI.org

[24] Samih M. Mostafa, S.Z. Rida and Safwat H. Hamad (October 2010), Finding time quantum of round robin cpu scheduling algorithm in general computing systems using integer programming, IJRRAS 5(1), PP. 64-71, Egypt.

[25] Saroj Hiranwal and Dr. K.C. Roy," Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice, (IJDE) International Journal of Data Engineering, Vol. 2, Issue 3, India. www.CSCjournals.org/CSC/manuscript/../IJDE-57.

[26] Shahram Saeidi and Hakimeh Alemi Baktash (2012, 10), Determining the Optimum Time Quantum Value in Round

Robin Process Scheduling Method, I.J. Information Technology and Computer Science, 67-73, Vol. 4, ISSN:2074-9007, Tabriz, Iran.

[27] Siddharth Tyagi, Sudheer Choudhary and Akshant Poonia (October 2012), Enhanced Priority Scheduling Algorithm to minimise Process Starvation, (IJETAE) International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 10, ISSN:2250-2459, India.

[28] Sourav Kumar Bhoi, Sanjaya Kumar Panda and Debashee Tarai (December 2012), Enhancing cpu performance using subcontrary mean dynamic round robin (SMDRR) scheduling algorithm, (JGRCS) Journal of Global Research in Computer Science, Vol. 2 , No. 12 , ISSN-2229-371X, PP. 17-21, India.

[29] Vishum Kumar Dhakad and Lokesh Sharma (May 2012), Performance analysis of round robin scheduling using adaptive approach based on smart time slice and comparison with SRR, (IJAET) International Journal of Advances in Engineering & Technology, Vol. 3, Issue 2, PP. 333- 339, ISSN:2231-1963, Jaipur, Rajasthan.