مقارنة أداء خوارزميات الجدولة في الزمن الحقيقي من حيث استهلاكها للطاقة عند العمل على معالج متعدد النوى

د. محمد حجازية*

يوسف نتيفة**

(تاريخ الإيداع 7 / 7 / 2021. قُبِل للنشر في 3/ 10 / 2021)

□ ملخّص □

أدى التقدم المتزايد والمستمر لتكنولوجيا المعلومات والاتصالات إلى الحاجة لتلبية متطلبات هذه التكنولوجيا والتي تشكل نظم الزمن الحقيقي النواة الأساسية لها وذلك بهدف تأمين جودة خدمة عالية تلائم طبيعة التطبيقات المنتشرة اليوم، لأن العامل الأساسي في نجاح أي نظام زمن حقيقي هو ضمان تنفيذ المهام قبل بلوغ القيد الزمني المرتبط بكل مهمة.

يتطلب الحصول على نظام زمن حقيقي عالي الأداء توفر خوارزميات جدولة من أجل ترتيب تنفيذ المهام وفقاً لأفضليات معينة بحيث تنهي جميع المهام التنفيذ من دون تجاوز القيد الزمني، وقد طرح الباحثون في هذا المجال العديد من الخوارزميات المختلفة من حيث مبدأ إسناد الأفضليات، وطبيعة منصة التشغيل، ونوع المهام التي تتعامل معها.

تتجه معظم الأبحاث في السنوات الأخيرة إلى تحسين استهلاك الطاقة في مختلف المجالات والتطبيقات البحثية، وبالنسبة لمجال الزمن الحقيقي فإن ذلك يعني على وجه الخصوص تحسين أداء خوارزميات الجدولة بحيث تؤمن الوظيفة المرجوة منها مع استهلاك أقل للطاقة.

تم في هذا البحث دراسة وتقييم مجموعة من خوارزميات الجدولة المختلفة من حيث استهلاكها للطاقة، وتم قياس الطاقة باستخدام حاسب يعمل بنظام تشغيل UBUNTU لكونه يحتوي على مجموعة من التوابع التي تتيح للمستخدم قياس مستويات استهلاك الطاقة عند تنفيذ مجموعة معينة من المهام، وقد تمت عملية محاكاة جدولة المهام بالزمن الحقيقي بالاعتماد على محاكي SIMSO وهو محاكي مفتوح المصدر ويتيح إمكانية محاكاة عدد كبير من خوارزميات الجدولة في الزمن الحقيقي، كما أنه متوافق مع بيئة نظام التشغيل UBUNTU.

الكلمات المفتاحية: الجدولة، المهام العشوائية، المعالج متعدد النوى، التوزع الاحتمالي.

* أستاذ – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.
mohammed.hejazieh2016@gmail.com

**طالب دكتوراه – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.
youssef.ntefeh@gmail.com

Comparing the Performance of Real-time Scheduling Algorithms in Terms of their Power Consumption when Working on a Multi-core Processor

Dr. Mohammed Hijazieh*
Youssef Ntefeh**

(Received 7 / 7 / 2021. Accepted 3 / 10 / 2021)

□ ABSTRACT □

The exponentially increasing progress of information and communication technology has led to the need to meet the requirements of this technology, of which real-time systems constitute the basic core, in order to ensure a high quality of service in line with the nature of applications spread today, because the main factor in the success of any real-time system is to ensure that tasks are executed before Reaching the time constraint associated with each task.

This led to the need for scheduling algorithms in order to arrange the execution of tasks according to certain preferences so that all tasks are completed without exceeding the time constraint. Researchers in this field have proposed many different algorithms in terms of the principle of attribution of preferences, the nature of the operating platform, and the type of tasks that deal with them.

Most of the research in recent years tends to improve energy consumption in various fields and research applications, and for the real-time field, this means in particular improving the performance of scheduling algorithms so that they provide the desired function with less energy consumption.

In this research, a group of different scheduling algorithms was studied and evaluated in terms of their energy consumption, and energy was measured using a computer running UBUNTU operating system because it contains a set of functions that allow the user to measure energy consumption levels when performing a certain set of tasks, and a scheduling simulation has been carried out Real-time tasks based on SIMSO, an open source simulator that allows simulation of a large number of scheduling algorithms in real time, and is compatible with the UBUNTU operating system environment.

Keywords: Scheduling, Sporadic Tasks, Multicore Processor, Probability Distribution.

* **Professor - Department of computer and automatic control Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Latakia, Syria. mohammed.hejazieh2016@gmail.com**
** **PhD Student in Department of computer and automatic control Engineering, Faculty of Mechanical and Electrical Engineering, Tishreen University, Latakia, Syria. Email: youssef.ntefeh@gmail.com**

مقدمة:

تلعب نظم الزمن الحقيقي دوراً أساسياً في العديد من المجالات العلمية والتطبيقية، وتُقسم إلى قسمين رئيسيين هما نظم الزمن الحقيقي المرنة Soft Real Time System، ونظم الزمن الحقيقي القاسية Hard Real Time System.

ويتمثل الفارق الأساسي بينهما في أن نظم الزمن الحقيقي المرنة تملك مرونة في التعامل مع المهام التي تتجاوز قيدها الزمني المرتبط بها، إذ لا يؤدي هذا التجاوز إلى نتائج خطيرة أو كارثية، مثل نظام المصادقة عند التسجيل على بريد الكتروني حيث يتم إرسال كود معين للتفعيل ويؤدي التأخير في إدخال الكود إلى فشل التسجيل مع إمكانية إعادة التسجيل مرة أخرى، في حين أن نظم الزمن الحقيقي القاسية تتسم بالتعامل بحزم مع هذا الموضوع لأن تجاوز المهمة لقيدها الزمني سوف يؤدي إلى فشل نظام الزمن الحقيقي والحصول على نتائج خطيرة أو كارثية، مثل أمر فتح الوسادة الهوائية في السيارة عند حدوث الاصطدام حيث يؤدي التأخير في تنفيذ مثل هذه المهام إلى موت السائق أحياناً أو إصابته إصابة بليغة.

أهمية البحث وأهدافه:

يهدف هذا البحث إلى إجراء دراسة تحليلية لخوارزميات الجدولة في نظم الزمن الحقيقي بهدف التوصل إلى الخوارزمية التي تحقق جدولة مجموعة عشوائية من المهام المختلفة من حيث النوع (دورية Periodic، عشوائية Sporadic، غير دورية Aperiodic) ومن حيث نسب استخدام المعالج Utilization (استخدام خفيف Light، عالي Heavy، زائد Overload) ومن حيث العدد، وذلك بأقل قدر ممكن من استهلاك الطاقة،

طرائق البحث ومواده:

سوف نستعرض لمحة موجزة عن نظام الزمن الحقيقي، وخوارزميات الجدولة وتصنيفاتها بالإضافة إلى الفوارق والميزات لكل منها، وبرنامج المحاكاة SIMSO المستخدم في هذا البحث.

1. نظام الزمن الحقيقي

هو عبارة عن أي نظام يرتبط تنفيذ المهام فيه بقيد زمني Deadline. بحيث يتوجب على خوارزمية الجدولة المستخدمة ترتيب تنفيذ هذه المهام بحيث تمنع تجاوز القيد الزمني، وتوجد ثلاثة أنواع من المهام في نظم الزمن الحقيقي هي:[6]

- المهام الدورية Periodic Tasks

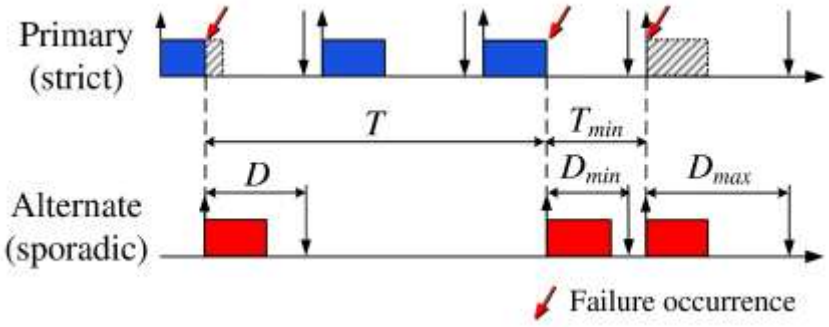
وهي عبارة عن مجموعة من المهام التي يتكرر حدوثها خلال فترات زمنية متتالية ومتساوية تسمى أدوار المهمة Tasks Periods، فإذا كان دور المهمة مساوياً للقيد الزمني الخاص بها عندئذ نطلق عليها المهام ذات القيد الزمني الضمني Implicit Deadline، أما إذا كان الدور مغايراً للقيد الزمني فتسمى بالمهام ذات القيد الزمني الصريح Explicit Deadline.

- المهام العشوائية Sporadic Tasks

هي المهام التي يتكرر حدوثها وفق أزمنة غير دورية لكن بشرط عدم إصدار نسختين من المهمة في نفس الوقت، أي يكون الفاصل الزمني بين كل إصدارين متتاليين للمهمة أكبر تماماً من الصفر.

- المهام غير الدورية Aperiodic Tasks

هي حالة عامة من المهام العشوائية حيث تملك نفس صفاتها مع فارق أنه يمكن ورود إصدارين متتاليين للمهمة في نفس الوقت.[7][3]



الشكل (1) الفرق بين المهام الدورية والعشوائية

2. خوارزميات الجدولة Scheduling Algorithms

هي عبارة عن مجموعة من الخوارزميات التي تهتم بترتيب تنفيذ المهام في نظم الزمن الحقيقي من خلال إسناد أفضلية معينة لكل مهمة بحيث تختلف كل خوارزمية عن الأخرى في المعيار المتبع لإسناد الأفضليات، بالإضافة إلى نوع المهام التي يمكن جدولتها عن طريق هذه الخوارزمية.

تم تصنيف خوارزميات الجدولة في نظم الزمن الحقيقي إلى عدة أصناف وفقاً لعدة معايير ونلخص فيما يلي تصنيفات خوارزميات الجدولة:[18][2]

a) الجدولة على منصّة عمل تعمل بمعالج وحيد Uniprocessor Scheduling

يُسند فيها رتل المهام إلى المعالج الوحيد الموجود في المنصة لكي تتم جدولته.

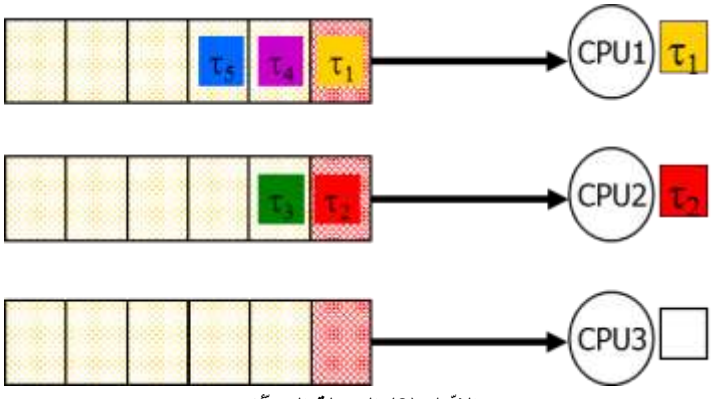
b) الجدولة على منصّة عمل بمعالجات متعدّدة Multiprocessor Scheduling

وفيها يُسند رتل المهام لكي ينفذ على كافّة المعالجات الموجودة في المنصّة.

وبدورها تُقسم الجدولة على منصّة عمل بمعالجات متعدّدة إلى قسمين هما:

a) الجدولة المجزّأة Partitioned Scheduling

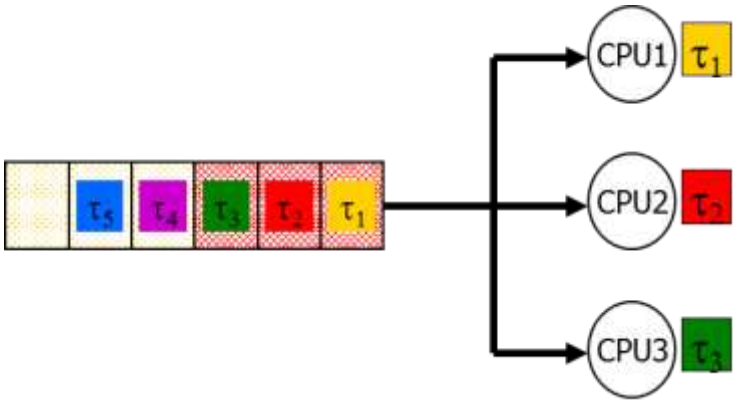
يُقسم رتل المهام إلى مجموعة من الأرتال الجزئيّة، ويُسند كلّ رتلٍ بدوره لكي يُجدول على معالجٍ وحيدٍ في المنصّة دون غيره، كأنّنا قد استبدلنا المنصّة ذات المعالجات المتعدّدة بمجموعة من المنصّات ذات المعالج الوحيد.



الشّكل (2) الجدولة المجزّأة

b) الجدولة الشاملة Global Scheduling

يُوزّع رتل المهام على كافّة معالجات المنصّة دفعةً واحدةً لكي ينفذ بحيث يتم اختيار مهمة ويخصص لها معالج غير مشغول في المنصة.



الشّكل (3) الجدولة الشاملة

ومن ناحية أخرى هناك تصنيف ثانٍ للجدولة من حيث كونها تتم قبل البدء بتشغيل النظام أو أثناء تشغيله، كما يلي:

a) الجدولة السّاكنة Static Scheduling:

حيث تعطى لكل مهمة موجود في النظام أفضلية تبقى ثابتة طوال فترة تشغيل النظام، حيث أن المهمة ذات الأفضلية الأعلى تنفذ عند ورودها بغض النظر عن وضع المهمة قيد التنفيذ.

ويشترط في هذه الجدولة أن تكون المهام الموجودة في النظام بالإضافة إلى أزمنة وصولها معروفة بشكل جيد من قبل المصمم حتى تُجدول بشكل أمثلي، لذلك تُطبق هذه الجدولة في الغالب على المهام الدورية.

تُجدول المهام في هذا الصّنف خلال زمن التّرجمة Compile Time، لذا فإنّ الأعباء Overheads النّاتجة أثناء تشغيل النّظام من حيث التبديل بين المهام وتفريغ ذاكرة الكاش والحصول على الموارد Resources تكون قليلة جدّاً مقارنة مع الصّنف الحركي، ويُعتبر هذا من ميزات هذا الصّنف، لكن عدم قدرة هذا الصّنف على جدولة المهام العشوائية Sporadic Tasks، والمهام غير الدورية Aperiodic Tasks يُشكل أهم المساوئ التي يعاني منها هذا الصّنف، والسّبب في ذلك صعوبة توقع زمن وصول مثل هذه المهام.

b) الجدولة الديناميكية Dynamic Scheduling:

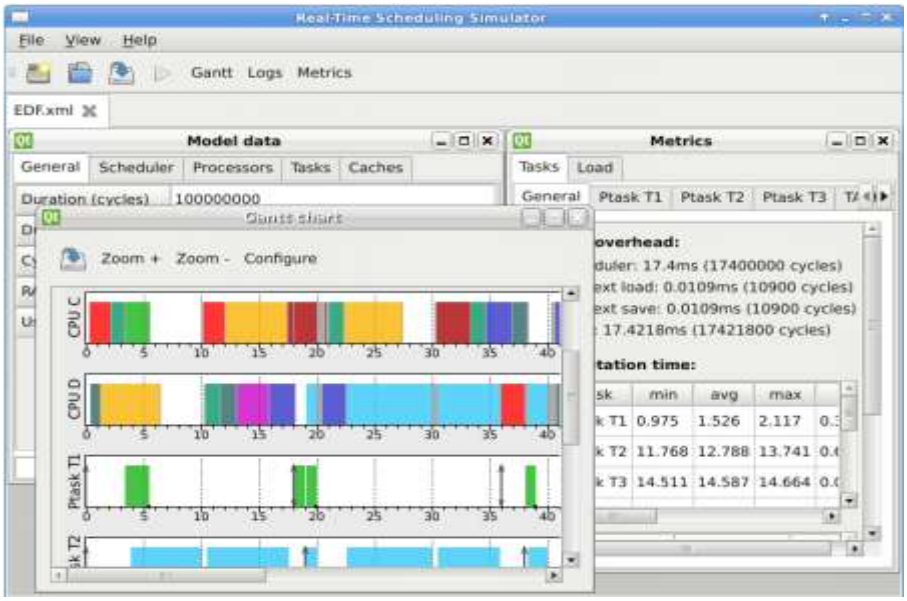
تمتلك المهام في هذا النظام أفضلية متغيرة حيث يتم إسناد الأفضليات إلى المهام أثناء التنفيذ وذلك وفقاً للقواعد التي تحددها خوارزمية الجدولة المختارة، الأمر الذي يعطينا نظام أسرع وأكثر فعاليّة كون أن زمن الترجمة يكون صغيراً مقارنة مع الصّنف السّابق (الستاتيكي) وبالتالي هناك سرعة في بدء تشغيل النّظام، ويُعدّ ذلك من أهمّ ميّزات هذا الصّنف بالإضافة إلى القدرة على جدولة المهام العشوائية، والمهام غير الدورية، بالمقابل هناك أعباء كثيرة تواجه النظام أثناء التشغيل من حيث التبديل بين المهام وتفريغ ذاكرة الكاش والحصول على الموارد، التي يمكن أن تتسبب في زمن تأخير أكبر بسبب كون الطلب على هذه الموارد يتم بشكل ديناميكي أثناء تشغيل النظام ويُعتبر ذلك من مساوئ هذا الصّنف.

وانطلاقاً من خصائص كلٍ من الصّنفين السابقين يمكننا القول إنّ اختيار أي منهما عند بناء نظام الزمن الحقيقي يعتمد على طبيعة المهام الموجودة في النظام فإذا كان الغالب على المهام أنّها دورية فإننا نختار الصنّف السّاكن، أمّا إذا كان الغالب على المهام أنّها ذات وصول عشوائي وغير دورية فإننا نختار الصّنف الحركي.[18][10][9] [19]

3. برنامج المحاكاة المستخدم في البحث

تم استخدام المحاكي SIMSO في هذا البحث لكونه يتمتع بالخصائص التالية:

1. مفتوح المصدر Open Source: أي يمكن تعديل أي ملف من المكتبات المضمنة في هذا المحاكي بما في ذلك الملفات التي تصف عمل خوارزميات الجدولة.
2. يؤمن واجهة مستخدم رسومية GUI (Graphical User Interface): وذلك من خلال مجموعة من الأزرار والخيارات التي تتيح ضبط إعدادات المحاكاة بشكل سهل ومرن.
3. يؤمن إحصائيات المحاكاة في الخرج بعد الانتهاء من عملية المحاكاة وهذه الإحصائيات تتضمن مثلاً أزمنة إطلاق المهمة، وكلفة عمليات تبديل السياق، وكلفة عمليات اتخاذ قرار الجدولة، وغيرها من الإحصائيات المفيدة.[5]
4. مكتوب بلغة الـ Python والتي تعد من اللغات البرمجية المتطورة، كما أنها تدعم البرمجة غرضية التوجه OOP (Object Oriented Programming).
5. السرعة في التنفيذ مقارنة مع محاكيات أخرى ضمن نفس المجال.[5]



الشكل (4) الواجهة الرسومية للمحاكي SIMSO

النتائج والمناقشة:

تمت دراسة ثلاثة سيناريوهات مختلفة تتضمن العمل وفق ثلاث خوارزميات هي:

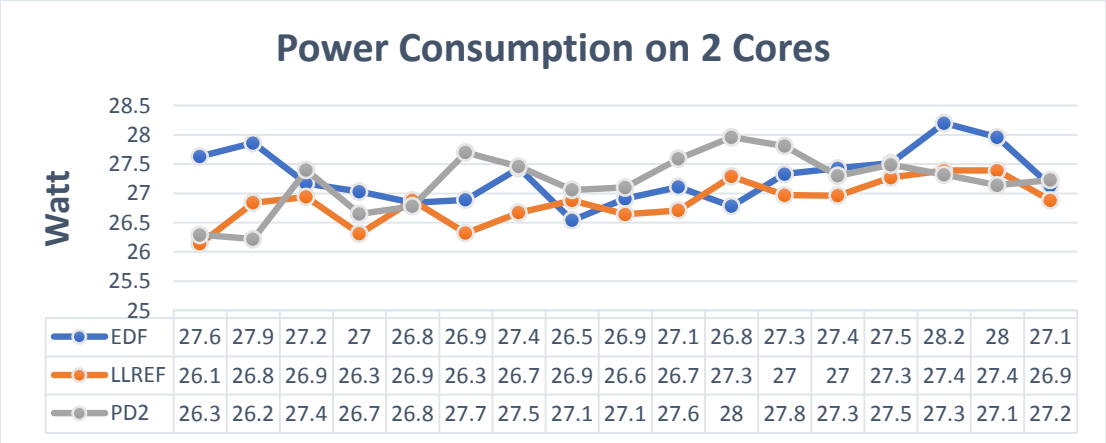
1. خوارزمية (Pseudo Deadline) PD2، وهي خوارزمية مقادة بالفترات الزمنية Time Slice Driven حيث يتم استدعاء المجدول من أجل ترتيب المهام بطريقة تضمن حصول المهمة التي تملك هامش زمني أقل للتنفيذ على أفضلية أعلى.
2. خوارزمية (Earliest Deadline First) EDF، وهي خوارزمية مقادة بالأحداث Event Driven، حيث يتم استدعاء المجدول عند ورود حدث إطلاق مهمة جديدة، وتُعطى المهمة ذات القيد الزمني الأقرب Nearest Deadline أفضلية أعلى.

3. خوارزمية LLREF (Largest Local Remaining Execution time First)، وهي خوارزمية مقادة بالأحداث، حيث يتم استدعاء المجدول عند ورود حدث إطلاق مهمة جديدة أو حدث إنهاء تنفيذ مهمة معينة أو حدث نفاذ الهامش الزمني لمهمة معينة، وتُعطى الخوارزمية التي تملك هامش زمني أقل أفضلية أعلى في التنفيذ.

يتم في السيناريو الأول تنفيذ المحاكاة على نواتين، وفي السيناريو الثاني على أربع نوى، وفي السيناريو الثالث على ثمان نوى، وذلك عند توليد مجموعات ذات عدد عشوائي من المهام والتي تملك أزمنة تنفيذ مختلفة ونسب استخدام Utilization مختلفة مع الأخذ بعين الاعتبار مقدار استهلاك الطاقة في كل سيناريو، ومع العلم أن المعالجات المستخدمة متجانسة.

1. سيناريو العمل على نواتين

تم في هذا السيناريو مقارنة أداء الخوارزميات الثلاثة من حيث استهلاك الطاقة وذلك عند جدولة نفس المجموعة من المهام المولدة بشكل عشوائي، في ظل العمل على نواتين.



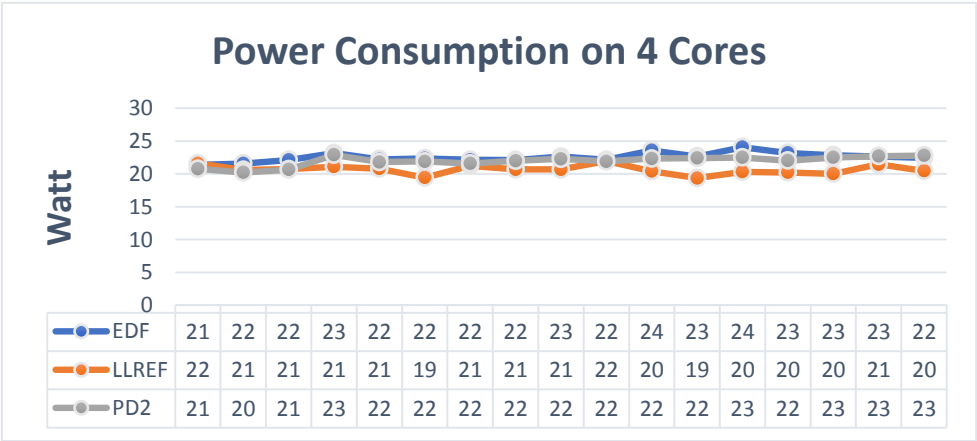
EDF	27.6	27.9	27.2	27	26.8	26.9	27.4	26.5	26.9	27.1	26.8	27.3	27.4	27.5	28.2	28	27.1
LLREF	26.1	26.8	26.9	26.3	26.9	26.3	26.7	26.9	26.6	26.7	27.3	27	27	27.3	27.4	27.4	26.9
PD2	26.3	26.2	27.4	26.7	26.8	27.7	27.5	27.1	27.1	27.6	28	27.8	27.3	27.5	27.3	27.1	27.2

الشكل (5) استهلاك الطاقة عند العمل على نواتين

نلاحظ من المخطط أن خوارزمية LLREF تتمتع بأقل قيم استهلاك للطاقة، في حين يكون أداء باقي الخوارزميتين متقاوتاً.

2. سيناريو العمل على أربع نوى

تم في هذا السيناريو مقارنة أداء الخوارزميات الثلاثة من حيث استهلاك الطاقة وذلك عند جدولة نفس المجموعة من المهام المولدة بشكل عشوائي، في ظل العمل على أربع نوى.



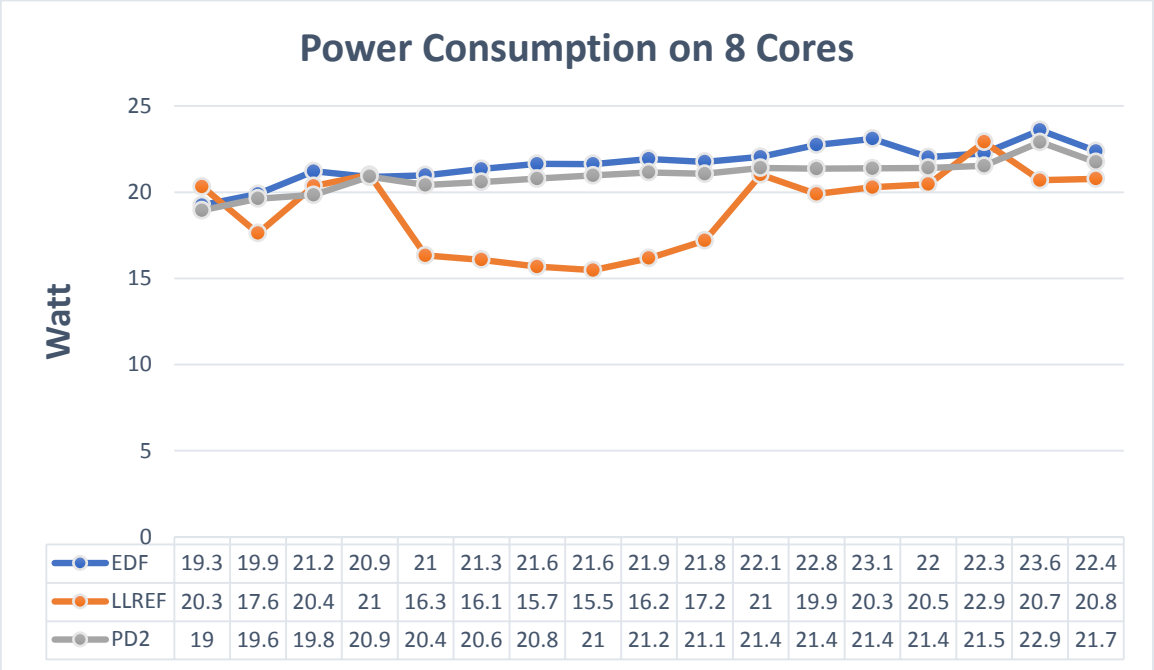
EDF	21	22	22	23	22	22	22	22	23	22	24	23	24	23	23	23	22
LLREF	22	21	21	21	21	19	21	21	21	22	20	19	20	20	20	21	20
PD2	21	20	21	23	22	22	22	22	22	22	22	22	23	22	23	23	23

الشكل (6) استهلاك الطاقة عند العمل على أربع نوى

نلاحظ من المخطط أن خوارزمية LLREF أقل استهلاكاً للطاقة وتليها خوارزمية PD2، وفي المرتبة الأخيرة خوارزمية EDF.

3. سيناريو العمل على ثماني نوى

تم في هذا السيناريو مقارنة أداء الخوارزميات الثلاثة من حيث استهلاك الطاقة وذلك عند جدولة نفس المجموعة من المهام المولدة بشكل عشوائي، في ظل العمل على ثماني نوى.



EDF	19.3	19.9	21.2	20.9	21	21.3	21.6	21.6	21.9	21.8	22.1	22.8	23.1	22	22.3	23.6	22.4
LLREF	20.3	17.6	20.4	21	16.3	16.1	15.7	15.5	16.2	17.2	21	19.9	20.3	20.5	22.9	20.7	20.8
PD2	19	19.6	19.8	20.9	20.4	20.6	20.8	21	21.2	21.1	21.4	21.4	21.4	21.4	21.5	22.9	21.7

الشكل (7) استهلاك الطاقة عند العمل على ثماني نوى

نلاحظ أن خوارزمية LLREF أصبحت أقل استهلاكاً للطاقة بفارق كبير عن الخوارزميتين الباقيتين، في حين تأتي خوارزمية PD2 في المرتبة الثانية، وتليها أخيراً خوارزمية EDF.

الاستنتاجات والتوصيات:

يمكن من خلال مراجعة النتائج الواردة في هذا البحث تلخيص مجموعة من الاستنتاجات في النقاط التالية:

1. تعتبر خوارزمية LLREF هي الأقل استهلاكاً للطاقة في كل السيناريوهات وبنسب تتراوح بين (10–35 %) ويرجع ذلك إلى العمليات المتبعة في هذه الخوارزمية من أجل تحديد سياسة الجدولة Scheduling Policy وإسناد الأفضليات للمهام، وهذه السياسة تتسم بقلة العمليات والتعقيد الحسابي والزمني المطلوب لاتخاذ القرار وهذا يؤدي إلى تنفيذ الجدولة بأقل قدر ممكن من استخدام وقت المعالجة والتوفير من استهلاك الطاقة.
2. نلاحظ أن أداء الخوارزمية LLREF يتحسن بفارق أكبر مع زيادة عدد النوى مما يعطي صورة أوضح عن مدى فعالية هذه الخوارزمية عند العمل على أنظمة تحتوي على عتاد مادي ضخم.
3. يؤدي التقليل من استهلاك الطاقة إلى إتاحة الإمكانية أمام نظام الزمن الحقيقي لتنفيذ عدد أكبر من المهام وهذه النقطة الإيجابية يمكن الوصول إليها بشكل واضح عند تطبيق الخوارزمية LLREF.
4. يساعد التقليل من الطاقة في عمل نظام الزمن الحقيقي لفترة زمنية أطول وخاصة إذا كانت منصة العمل متنقلة Mobile Platform مما يرفع من كفاءة وموثوقية نظام الزمن الحقيقي.

References:

1. Irraivan Elamvazuthi, "**Electrical power consumption monitoring using a real-time system**", Universiti Teknologi PETRONAS, (2021)
2. Radhakrishna Naik at el, "**Periodic and Aperiodic Real - Time Task Scheduling Algorithms Simulator** ", International Journal of Pure and Applied Mathematics, Volume 118 No. 20 ,2681-2687 (2018).
3. Hyeongboo BAEK, "**Real-Time Scheduling for Preventing Information Leakage with Preemption Overheads** ", Advances in Electrical and Computer Engineering, Volume 17, Number 2, (2017).
4. Ankur Jain , **" Multishare Task Scheduling Algorithm For Real Time Micro-controller Based Application " ,** Mechatronics and Applications: An International Journal (MECHATROJ), Vol. 1, No.1, (2015).
5. Schoeberl, M., Pezzarossa, "**A multicore processor for time-critical applications** ", Technical University of Denmark , (2018).
6. Charu Rani, Mrs. Manju Godara , **" Real Time System Scheduling Algorithms & Fault Tolerance "** , International Journal of Advanced Research in Computer and Communication Engineering , Vol. 4, Issue 7, July (2015).
7. Sri Raj Pradhan, Sital Sharma, Debanjan Konar, **"A Comparative Study on Dynamic Scheduling of Real-Time Tasks in Multiprocessor System using Genetic Algorithms "** , International Journal of Computer Applications , Volume 120 – No.20, June (2015).
8. Christine Rochange, **"Parallel Real-Time Tasks, as Viewed by WCET Analysis and Task Scheduling Approaches "** , University of Toulouse, Toulouse, France (2016).
9. Fernanda F. Peronaglio and Aleardo Manacero, "**MODELING REAL-TIME SCHEDULERS FOR USE IN SIMULATIONS THROUGH A GRAPHICAL INTERFACE",** Dept of Computer Science and Statistics São Paulo State University – UNESP, (2017).
10. Karthik S. Lakshmanan, **" Scheduling and synchronization for Multicore Real Time Systems ",** Carnegie Mellon University, Pittsburgh, PA, (2011).
11. Manar Qamheih, **"Scheduling of Parallel Real-time DAG Tasks on Multiprocessor Systems",** University of Paris-est (2015).
12. Dan McNulty, Lena Olson, Markus Peloquin , "**A Comparison of Scheduling Algorithms for Multiprocessors**" , University of Maryland (2010).
13. Greg Levin , "**DP-FAIR: A Simple Model for Understanding Optimal Multiprocessor Scheduling**" International Journal of Computer Applications (0975 – 8887) Volume 110 – No. 6 (2009).
14. Kushal Anjaria, Arun Mishra, "**Thread scheduling using ant colony optimization: An intelligent scheduling approach towards minimal information leakage**", Department of Computer Science & Engineering, DIAT, Pune, India, (2017).
15. Gokul Sidarth Thirunavukkarasu*****, and Ragil Krishna, "**Scheduling Algorithm for Real-Time Embedded Control Systems Using Arduino Board**", Deakin University, School of Engineering, Waurn ponds, Australia, (2017).
16. Amjad Mahmood , Salman A. Khan, "**Energy-Aware Real-Time Task Scheduling in Multiprocessor Systems Using a Hybrid Genetic Algorithm**", University of Bahrain, (2017).

17. Linlin Tang*a*, Kaiqiang Ma, Zuohua Li, "**A New Scheduling Algorithm Based on Ant Colony Algorithm and Cloud Load Balancing**", Harbin Institute of Technology Shenzhen Graduate School Shenzhen, China, (2016).

18. Khaldoun Faqi, Dr. Muhammad Hijazia**, "Improving the performance of real-time systems by optimizing task scheduling on a multi-core processor",** PhD thesis in the Department of Computer Engineering and Automatic Control, Tishreen University, Lattakia, Syria (2019).

19. Rola Marisha, Dr. Muhammad Hijazia**, "Design and Modeling of an Adaptive Scheduling Algorithm for Real-Time Decision Making",** PhD Thesis, Department of Computer Engineering and Automatic Control, Tishreen University, Lattakia, Syria (2015).