مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (3) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (3) 2016

نمذجة خوارزميات الجدولة للمهام الدورية وتحليل أدائها في أنظمة الزمن الحقيقي متعددة المعالجات

الدكتور محمد حجازية*
رلى مريشة**

(تاريخ الإيداع 6 / 9 / 2015. قُبِل للنشر في 20/ 6 / 2016)

□ ملخّص □

يقدم البحث نمذجة وتحليل أداء عدد من خوارزميات الجدولة في أنظمة الزمن الحقيقي متعددة المعالجات. حيث تم تحليل أداء كل من الخوارزميات الثلاث: خوارزمية الجدولة بالزمن الحرج الأقصر أولاً EDF ، و خوارزمية الجدولة بالزمن الأقل خمولاً أولاً LLF ، وخوارزمية الجدولة بالزمن الحرج أولاً عند الخمول الصفري EDZL . شملت هذه الدراسة جدولة مهام دورية ذات قيود زمنية مساوية لدورها ، ومستقلة، و قابلة للمقاطعة على عدة معالجات متطابقة . تمت مقارنة الخوارزميات الثلاث من ناحية الحمل على المعالج (مشغولية المعالجات)، ومن ناحية عدد الهجرات، وعدد المقاطعات، وعدد المرات التي لم تنجح فيها هذه الخوارزميات في تحقيق الحدود الزمنية للمهام، حيث يعتبر الأخير أهم معيار من معايير عملية الجدولة في الزمن الحقيقي. كما تضمنت الدراسة جدولة مجموعات متزايدة من المهام الدورية تبدأ من 4 مهام لتصل حتى 64 مهمة ، وذلك لدراسة تأثير ازدياد عدد المهام والمعالجات على أداء خوارزميات الجدولة، وكنتيجة يقدم البحث نقاط القوة والضعف في أداء هذه الخوارزميات ويقترح لكل خوارزمية –حسب نقاط القوة في أدائها– نوع منظومة الزمن الحقيقي التي من الأفضل تطبيقها فيها.

الكلمات المفتاحية: خوارزميات الجدولة ، المهام الدورية ، أنظمة الزمن الحقيقي ، الأنظمة متعددة المعالجات

* أستاذ مساعد – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.
** قائم بالأعمال – قسم هندسة الحاسبات والتحكم الآلي– كلية الهندسة الميكانيكية والكهربائية– جامعة تشرين– اللاذقية– سورية.

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (3) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (3) 2016

Modeling the Scheduling Algorithms of Periodic Tasks and Analyzing its Performance on Real-Time Multiprocessor Systems

Dr. Mohammed Hijazieh*
Rula Mreisheh**

(Received 6 / 9 / 2015. Accepted 20 / 6 / 2016)

□ ABSTRACT □

The research presents molding and analytical study of several scheduling algorithms types in real-time multiprocessor systems. The performance of three scheduling algorithms have been analyzed : Earliest Deadline First Scheduling (EDF) , Least Laxity First Scheduling (LLF), and Earliest Deadline First until Zero Laxity Scheduling (EDZL). This paper considers the scheduling of n periodic, independed, and preempted tasks with implicit deadlines on a platform of m homogenous multiprocessor. It has compared in terms of the load on the processor (processor's busyness) , the number of migrations, and the number of preemptions and the number of times in which these algorithms did not succeed in achieving the time limits for tasks where the latter is considered the most important criterion in real time scheduling. It also considers scheduling growing task sets of periodic tasks starting from 4 task set up to 64 task set, in order to study the effect of increasing the number of tasks and processors also on the performance of the scheduling algorithms. As a result of research, the strengths and weaknesses in the performance of these three algorithms have presented. It is proposed the best type of real-time system to apply each algorithm according to the strengths of its performance.

Key Words: scheduling algorithms, periodic tasks, real-time systems, multiprocessor systems.

*Associate Professor, Department of computer and automatic control Engineering, Faculty of Mechanical and electrical Engineering, Tishreen University, Lattakia, Syria.

**Academic Assistant, Department of computer and automatic control Engineering, Faculty of Mechanical and electrical Engineering, Tishreen University, Lattakia, Syria.

مقدمة:

الهدف الأساسي في أنظمة الزمن الحقيقي ليس فقط الحصول على نتائج صحيحة وإنما يجب أن تعطى هذه النتائج ضمن الزمن المحدد . يتألف نظام الزمن الحقيقي بشكل عام من نظام تحكم (حاسب) ونظام متحكم به (البيئة) حيث يتفاعل نظام التحكم مع البيئة بالاعتماد على المعلومات المتوفرة عنها والتي تؤمنها له حساسات على شكل قراءات قد تكون دورية يفصل بينها فاصل زمني أو قد تكون لادورية تحدث بشكل غير منتظم أي (شبه دوري) أو قد تحصل عمليات بشكل مفاجئ. وبالتالي تعتمد قدرة الحاسب على الاستجابة لهذه العمليات أو المهام على أداء الحسابات الضرورية اللازمة ضمن الزمن المحدد لكل مهمة من هذه ال مهام. تعرف **المهمة Task**: بأنها وحدة التنفيذ الأساسية الخاصة التي تتم عليها عملية الجدولة حيث يتم تخصصيها في وحدة المعالجة المركزية (CPU) استناداً إلى خوارزميات الجدولة [1,2]. و تحدد **الخصائص الزمنية للمهمة** كالتالي:

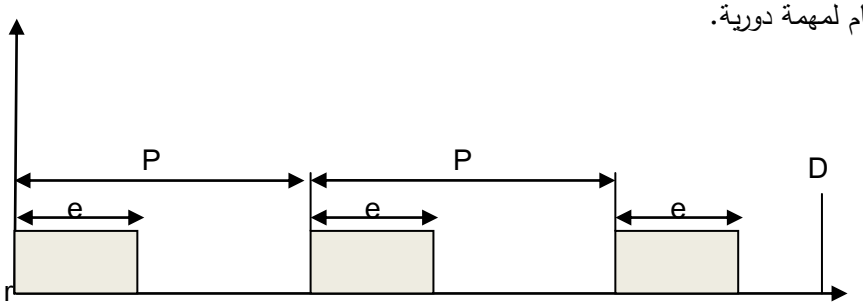
1 زمن انطلاق المهمة Release time (r) : هو زمن بداية تنفيذ طلبات المهمة.

2 زمن تنفيذ المهمة الأعظمي Worst case execution time(e) : هو الزمن الذي يوضع فيه المعالج تحت تصرف المهمة و يرمز له أيضاً (WCET).

3 زمن الموعد الحرج النسبي Deadline (D) : هو الزمن الحدي المسموح به إنجاز المهمة.

4 دور المهمة Task Period (P): وذلك بالنسبة للمهمات الدورية المتوافقة .

5 عامل الاسترخاء النسبي Laxity (L): هو زمن التأخير في بدء تنفيذ المهمة (L=D−e)، ويبين الشكل (1) الشكل العام لمهمة دورية.



الشكل (1) المهمة الدورية على محور الزمن

وفي حال ورود عدة مهام بنفس الوقت يحتاج الحاسب عندها لجدولة الحسابات للعمليات الواردة ليتمكن من الاستجابة لكل مهمة ضمن الوقت المخصص لها ، وفي حال عدم تمكن الحاسب من ذلك قد يؤدي هذا إلى عدة عواقب تتفاوت في خطورتها حسب نوع المهمة. فمنظومة الزمن الحقيقي اللينة soft real time system تراعي القيود الزمنية المفروضة عليها ولكن إذا أخفقت استمر النظام بالعمل لكن بدون الفائدة المرجوة ولا يؤدي بالضرورة إلى نتائج كارثية ، كلما أنهت المهمة عملها بعد الموعد النهائي كلما كانت النتائج أسوء ومثال ذلك، أجهزة الصراف الآلي ATM [1,2].

أما منظومة الزمن الحقيقي الصلبة hard real time system هو نظام يستخدم في إدارة المهمات الحساسة التي تتطلب دقة متناهية بالعمل يجب أن تستجيب هذه المنظومة في المدة الزمنية المطلوبة حتماً، وإلا قد يؤدي ذلك إلى نتائج كارثية أو خسائر مادية وكمثال نظام التحكم في الطائرة، ونظام الكبح المضاد للإقفال ABS في السيارات.

ومن هنا جاءت أهمية **المجدول** ضمن الحاسب الذي يقوم باستخدام خوارزمية جدولة معينة لتلبية وتنفيذ عمل جميع هذه المهام خلال زمن التنفيذ المحدد لها أي يتم مراعاة القيود الزمنية للمهام حيث يتم في نهاية عملية الجدولة اتخاذ قرار ببدء تنفيذ أحد هذه المهام في **الأنظمة وحيدة المعالج** وهي الأنظمة التي يتم فيها تنفيذ مهمة واحدة من رتل المهام وتبقى المهام الأخرى منتظرة حتى الانتهاء من التنفيذ ، أو يعمل المجدول على البدء بتخصيص عدة مهام على عدة معالجات في **الأنظمة المتعددة المعالجات** وهي الأنظمة التي يتم فيها تنفيذ أكثر من مهمة في الوقت نفسه وذلك من خلال توزيع المهام على المعالجات.

حيث تقسم المعالجات إلى نوعين: **معالجات متطابقة** (Homogenous):المعالجات لها نفس الخصائص تماماً مثل تطابق عدد المسجلات وسرعة المعالجة **ومعالجات متغايرة** (Heterogeneous):لا تكون للمعالجات نفس الخصائص تماماً بل تختلف بعدد المسجلات وسرعة المعالجة والسعة [2,3].

وتقسم **أنواع المهام في الزمن الحقيقي** إلى:

– مهام معتمدة / مستقلة (In-depended/ depended Tasks): عندما تحتاج المهمة إلى معلومات من مهمة أخرى حتى تتمكن من إتمام عملها تعتبر معتمدة أما المستقلة فلا تحتاج إلى معلومات من مهام أخرى.

– مهام شفعية/لاشفعية (preemptive, non-preemptive Tasks): هي مهام قابلة للمقاطعة او غير قابلة للمقاطعة.

– مهام الأولويات الثابتة/ الديناميكية (Fixed/ Dynamic priority Tasks):بعض المهام تكون ذات أولوية ثابتة طوال فترة العمل وبعضها الأخر تتغير أولويتها خلال سياق التنفيذ.

– مهام دورية / لادورية/ شبه دورية (Periodic/ Aperiodic /Speriodic Tasks) [3,4].

يعنى هذا البحث بجدولة المهام الدورية ذات قيد زمني مساوي لدورها ، المستقلة ، قابلة للمقاطعة على عدة معالجات متطابقة باستخدام عدة خوارزميات جدولة هي : EDF، LLF ، EDZL .

أهمية البحث وأهدافه:

هدف البحث هو نمذجة وتحليل أداء الخوارزميات الأكثر شهرة واستخداماً وهي EDF ، LLF ، و EDZL لتحديد الخوارزمية الأفضل من ناحية تحقيق القيود الزمنية للمهام الدورية ، المستقلة ، و القابلة للمقاطعة على عدة معالجات متطابقة، و هو أهم معيار من معايير الأداء في الزمن الحقيقي لأن الإخفاق في تحقيقه إذا لم يؤدي إلى نتائج كارثية قد يؤدي إلى فقدان المنظومة للهدف المرجو منها، ومن ناحية الحمل على المعالجات ومن ناحية تقليل عدد المقاطعات التي قد تحدث للمهام أثناء عملية الجدولة بسبب ورود مهمة ذات أولوية أعلى تحتاج العمل على نفس المعالج ،عندها تبقى المهام منتظرة حتى تنتهي المهمة ذات الأولوية الأعلى من عملها لتكمل من جديد على نفس المعالج ، ومن ناحية عدد الهجرات فقد يحدث أن تتم مقاطعة مهمة ما على أحد المعالجات و تستمر بعملها ولكن على معالج آخر.

إن هذه المقاطعات وهذه الهجرات تزيد من كلفة الجدولة وتزيد أيضاً من وقت الجدولة على المعالجات ولذلك تعمل الكثير من الأبحاث على دراسة أداء الخوارزميات لتحديد أيها الأفضل من ناحية عدد المقاطعات و عدد الهجرات

وتدرس أيضاً الحمل على المعالج و تحقيق القيود الزمنية للمهام. ومن هنا تأتي أهمية البحث بتقييم أداء هذه الخوارزميات بالنسبة لبارامترات المقارنة المدروسة وذلك لمعرفة الخوارزمية الأفضل استخداماً في نظم الزمن الحقيقي.

طرائق البحث ومواده:

عند البدء بدراسة خوارزميات الجدولة يجب معرفة القواعد التي تخضع لها عملية جدولة المهام ، حيث يتم حساب إنجازية utilization كل مهمة Task (T) كما يلي :

$$U_i = e_i / p_i$$

حيث : i=1,2 ,3.....n تمثل رقم المهمة و e_i هو زمن تنفيذ المهمة الأعظمي وp_i هو دور المهمة .

تكون مجموعات المهام الدورية قابلة للجدولة على عدة معالجات متطابقة إذا وفقط إذا:

$$U_{max}(T) <= 1$$

$$U_{sum}(T) <= m$$

أي أن تكون الإنجازية utilization الأعظمية للمهمة الواحدة أقل أو تساوي الواحد وأن يكون مجموع الإنجازيات utilization لكل المهام الداخلة إلى الجدولة أصغر أو تساوي عدد المعالجات m [5,6].

3-1معايير قياس أداء خوارزميات الجدولة:

توجد عدة معايير تحدد كيفية قياس أداء خوارزميات الجدولة وحساب فعاليتها، ومن أهم هذه المعايير:

- عدد مرات تجاوز القيد الزمني Deadline Misses: وهي عدد المرات التي أخفقت فيها الخوارزمية في تنفيذ المهمة قبل الوقت النهائي المحدد لها ، كلما كان أقل كلما كانت الخوارزمية أفضل.
- عدد المقاطعات Preemptions: هو عدد المرات التي تقوم مهمة ذات أولوية أعلى بمقاطعة عمل مهمة أخرى ذات أولوية أقل منها يتم تنفيذها على المعالج. عندها تبقى المهمة التي تمت مقاطعتها في حالة انتظار حتى تنتهي ذات الأولوية الأعلى من التنفيذ لتعود مجدداً وتستكمل عملها على نفس المعالج. تسبب المقاطعات زيادة في كلفة وزمن عملية الجدولة وبالتالي كلما كانت أقل كلما كانت الخوارزمية أفضل.
- عدد الهجرات Migrations : عدد المرات التي تقوم مهمة ذات أولوية أعلى بمقاطعة عمل مهمة أخرى ذات أولوية أقل منها ويتم تنفيذها على المعالج، بعدها تُكمل المهمة التي تمت مقاطعتها عملها على معالج آخر غير مشغول.هذا ما يدعى بعملية تبديل السياق context switch وهي تسبب أيضاً زيادة في كلفة وزمن الجدولة بسبب عملية المقاطعة.
- الحمل على المعالجات (مشغولية المعالجات) CPU Total Load: و هي مقدار الزمن الذي تكون فيه المعالجات قيد العمل وكلما كان أكبر كلما كان أفضل بمعنى أن الخوارزمية تستثمر المعالج بشكل جيد بدون مرور فترة زمنية يكون فيها المعالج عاطل عن العمل [4,5,6].

3-2خوارزميات الجدولة المستخدمة في البحث:

1 خوارزمية الزمن الحرج الأقصر أولاً Earliest Deadline First Scheduling (EDF):

هي خوارزمية مقادة بالأولويات وفيها الأولوية الأعلى تعطى للمهمة التي تملك الزمن الحدي الأقل [7].

2 خوارزمية الزمن الأقل خمولاً أولاً Least Laxity First Scheduling (LLF):

تعمل هذه الخوارزمية على إعطاء أعلى أولوية للمهمة النشطة والتي تحوي أقل زمن متبقي(الزمن المتبقي هو الفرق بين الزمن الباقي من تنفيذ المهمة والزمن الحدي للمهمة) ثم يتم تنفيذ المهمة ذات الأولوية الأعلى. في أثناء تنفيذ مهمة ما باستعمال هذه الخوارزمية، قد تخفف أولويتها إذا وردت مهمة أخرى لها زمن متبقي أقل من زمن هذه المهمة لهذا تعتبر خوارزمية جدولة ديناميكية. نصادف مشكلة في حال وصول مهمتين لهما نفس الزمن المتبقي حيث سيتم تنفيذ قسم من المهمة الأولى ثم تقاطعها الثانية لينفذ قسم منها وبالعكس .لذلك سيحصل العديد من تبديلات السياق أثناء زمن حياة العملية .[8]

3 خوارزمية الزمن الحرج أولاً عند الخمول الصفري Earliest Deadline First until Zero Laxity Scheduling (EDZL):

هذه الخوارزمية عبارة عن دمج عمل الخوارزميتين السابقتين،حيث تعمل على إعطاء الأولوية الأعلى للمهمة التي تملك الزمن الحدي الأقل ولكن إذا وردت مهمة أخرى لها زمن متبقي صفري عندها تعطى الأولوية الأعلى، ولهذا أيضاً تعتبر خوارزمية جدولة ديناميكية [9] .

3-3 برنامج المحاكاة المستخدم في البحث:

تمت عملية المحاكاة ورصد النتائج باستخدام برنامج حديث في هذا المجال تم العمل به أول مرة عام 2013 يدعى (SimSo) وهو اختصار Simulation of Real-Time Multiprocessor Scheduling with Overheads [10].

النتائج والمناقشة:

تمت الدراسة بجدولة عدد متزايد من المهام الدورية ،المستقلة، القابلة للمقاطعة (n مهمة) على عدد متزايد أيضاً من المعالجات المتطابقة (m معالج) بحيث :

n= 4, 8, 16, 32, 64 مهمة على m= 2, 4, 8, 16, 32 معالج، أي (n=2m) عدد المهام ضعف عدد المعالجات وذلك لدراسة أثر ازدياد المهام والمعالجات على أداء خوارزميات الجدولة المدروسة. إن اختيار عدد المهام ضعف عدد المعالجات لا يعني بالضرورة أن يكون لكل معالج مهمتين، لأن المهام دورية وبالتالي من الممكن أن ترد مهمة ذات تواتر دوري كبير فتشغل معالج واحداً وتتوزع باقي المهام على باقي المعالجات. وقد تم اختيار هذه النسبة في عدة دراسات مرجعية حديثة في هذا المجال مثل الدراسة [5].

يعمل برنامج المحاكاة على توليد مجموعات المهام الدورية المطلوبة بحيث يكون $U_{max}(T) <= 1$ ، و يختار الباحث قيمة U_{sum} بحيث تكون $U_{sum}(T) <= m$. تم اختيار $\mathbf{U_{sum}(T) = m}$ لكافة عمليات الجدولة أي مجموع إنجازيات كافة المهام يساوي الحد الأعلى المسموح به وهو عدد المعالجات.

باعتبار المهام الدورية متكررة بدور ثابت ، تم تحديد قيم دور المهام period ضمن المجال [50 1] وتسجيل النتائج الظاهرة عبر نافذة مراقبة observation window مجالها بين [100 0].

4-1 جدولة أربع مهام دورية (n=4) على معالجين متطابقين(m=2):

مثال عن جدولة أربع مهام دورية على معالجين متطابقين باستخدام خوارزميات الجدولة : حيث يتم في البداية توليد مجموعات مهام عددها مساوي لـ 4 و إنجازية كل مهمة منها أصغر من الواحد و مجموع إنجازيات المهام الأربع مساو لعدد المعالجات أي 2، كما في الشكل (2).

id	Name	Task type	bort on mi	Date	Period (ms)	ct. da	Deadline (ms)	WCET (ms)
1	Task 1	Periodic	No	0.0	30.0	-	30.0	24.791479
2	Task 2	Periodic	No	0.0	49.0	-	49.0	23.826848
3	Task 3	Periodic	No	0.0	21.0	-	21.0	9.186933
4	Task 4	Periodic	No	0.0	7.0	-	7.0	1.749175

الشكل (2) الخصائص الزمنية لأربعة مهام دورية في برنامج SimSo

حساب انجازية كل مهمة ومجموع إنجازيات المهام:

$U_{max}(T1) = 24.791/30 = 0.826 < 1$

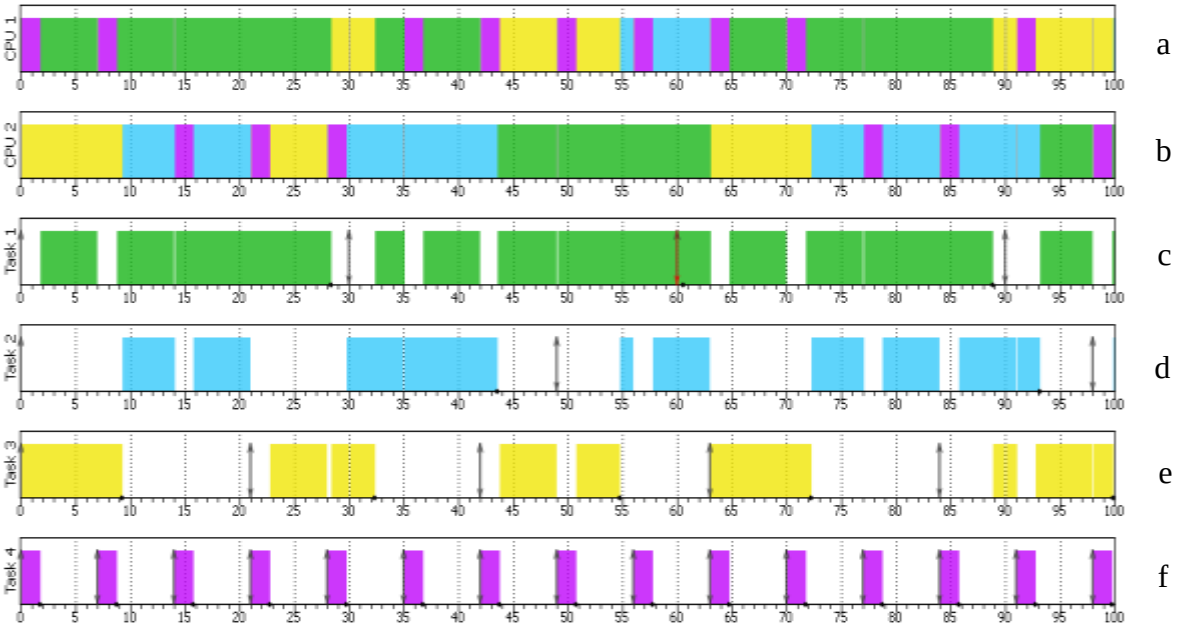
$U_{max}(T2) = 23.826/49 = 0.486 < 1$

$U_{max}(T3) = 9.186/21 = 0.437 < 1$

$U_{max}(T4) = 1.749/7 = 0.249 < 1$

$U_{sum}(T) = U_{max}(T1) + U_{max}(T2) + U_{max}(T3) + U_{max}(T4) = m = 2$

باختيار خوارزمية الجدولة الزمن الحرج الأقصر أولاً (EDF) سيتم توزيع المهمة Task4 و المهمة Task3 بداية على المعالجين على اعتبار أن لهما الزمن الحرج الأقصر كما يبين الشكل(3):



الشكل (3) جدولة أربع مهام دورية على معالجين باستخدام خوارزمية الجدولة EDF

نلاحظ من الشكل (3-a) و (3-b) أن المهمة Task1 قد تمت مقاطعتها 7 مرات وتم تهجيرها من معالج إلى آخر مرتين ولم تحقق القيد الزمني لها عند اللحظة 60، أما المهمة Task4 لم تتم مقاطعتها أو تهجيرها أثناء تنفيذها ولم تفقد أي قيد زمني خلال عملية الجدولة (الزمن الحرج المساوي لدور المهمة Task4 يتكرر كل 7).

يعطي البرنامج هذه القيم لكل مهمة على حدا، ويعطي أيضاً قيمة الحمل لكل معالج على حدا أيضاً فمثلاً في هذا المثال يعطي مقدار الحمل على المعالج الأول 1.0000 أي 100% و مقدار الحمل على المعالج الثاني 0.9998 أي 99.98% وبالتالي الحمل على المعالجين هو المتوسط الحسابي للقيمتين السابقتين أي 99.99%. تم العمل على جمع عدد الهجرات والمقاطعات وعدد المرات التي تم تجاوز القيد الزمني فيها لجميع المهام ومقدار الحمل على المعالجات في كل مرة تمت فيها عملية الجدولة باستخدام خوارزمية EDF.

باختيار خوارزمية الجدولة الأقل خمولاً أولاً (LLF) لجدولة نفس المهام سيتم توزيع المهمة Task1 و المهمة Task4 بداية على المعالجين على اعتبار أن لهما الزمن الخمول الأقل (هو الفرق بين الزمن الباقي من تنفيذ المهمة والزمن الحدي للمهمة) كما يبين الشكل (4) حيث:

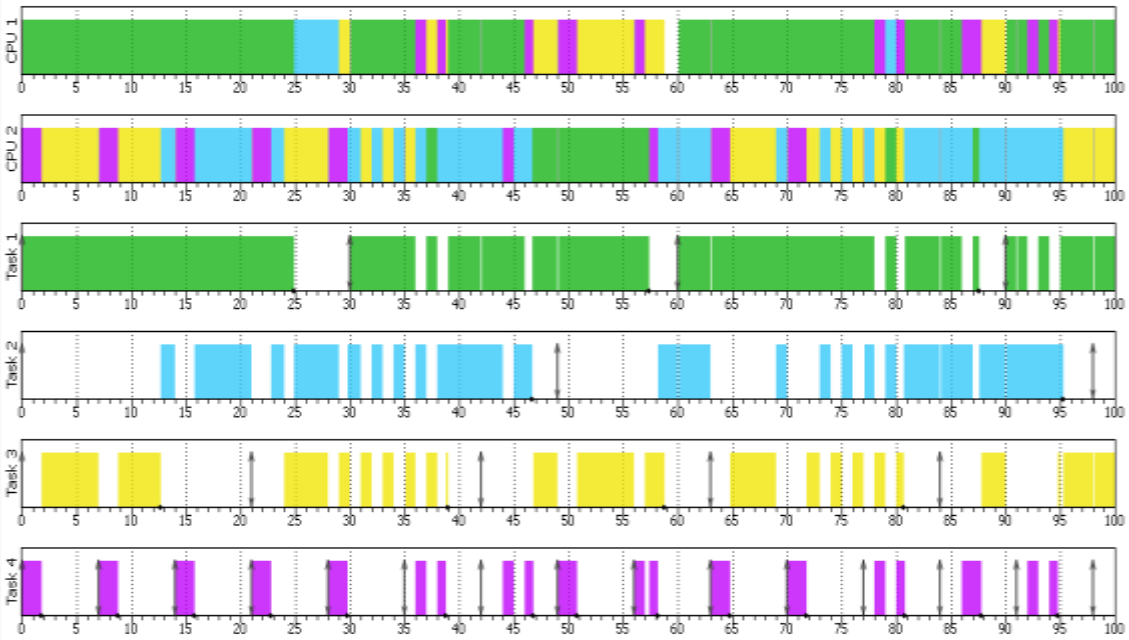
$$L(T1) = 30-24.79 = 5.21$$

$$L(T2) = 49-23.82 = 25.18$$

$$L(T3) = 21-9.1 = 11.9$$

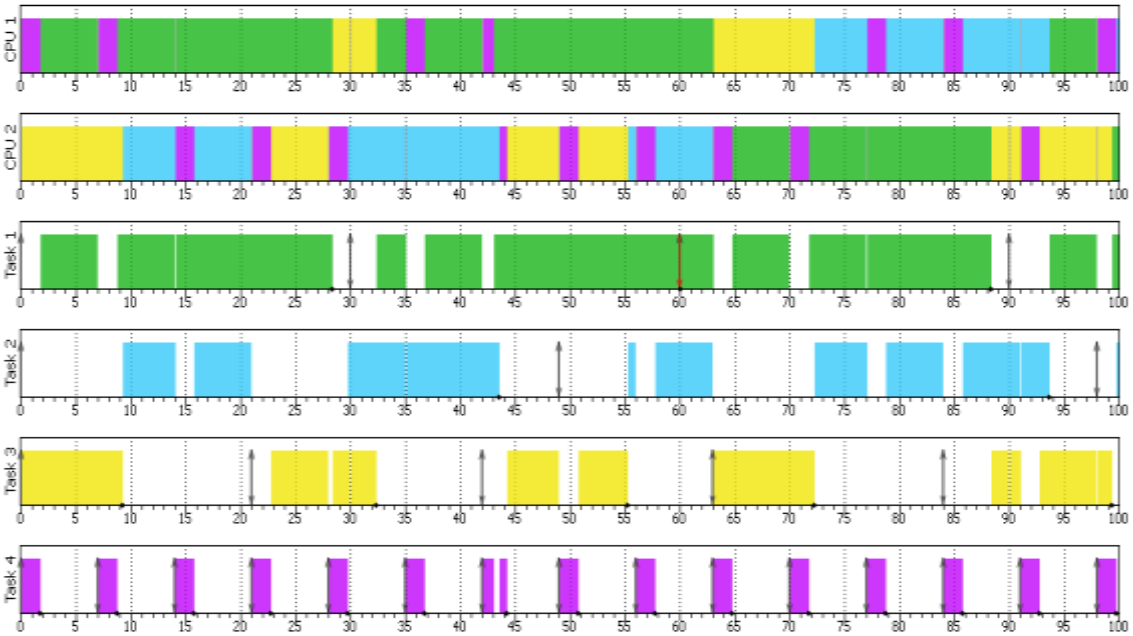
$$L(T4) = 7-1.749 = 5.25 \quad ==\rightarrow \quad L(T1) < L(T4) < L(T3) < L(T2)$$

نلاحظ على سبيل المثال أن المهمة Task4 قد تمت مقاطعتها 5 مرات وتم تهجيرها من معالج إلى آخر مرتين ولم تفقد القيد الزمني لها عند أي لحظة.



الشكل (4) جدولة أربع مهام دورية على معالجين باستخدام خوارزمية الجدولة LLF

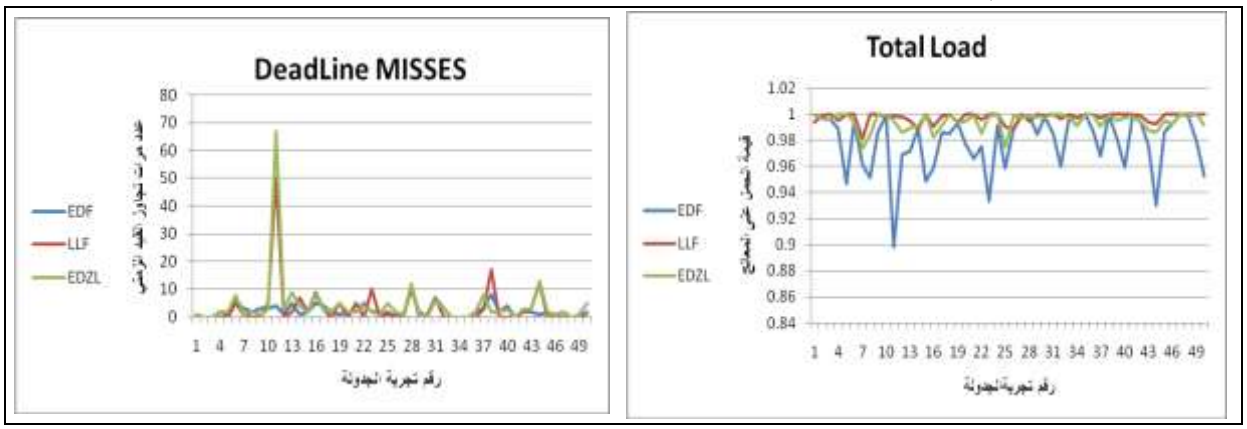
باختيار خوارزمية الجدولة الزمن الحرج الأقصر أولاً عند الخمول الصفري (EDZL) لجدولة نفس المهام سيتم توزيع المهمة Task4 و المهمة Task3 بداية على المعالجين على اعتبار أن لهما الزمن الحرج الأقصر كما يبين الشكل (5):

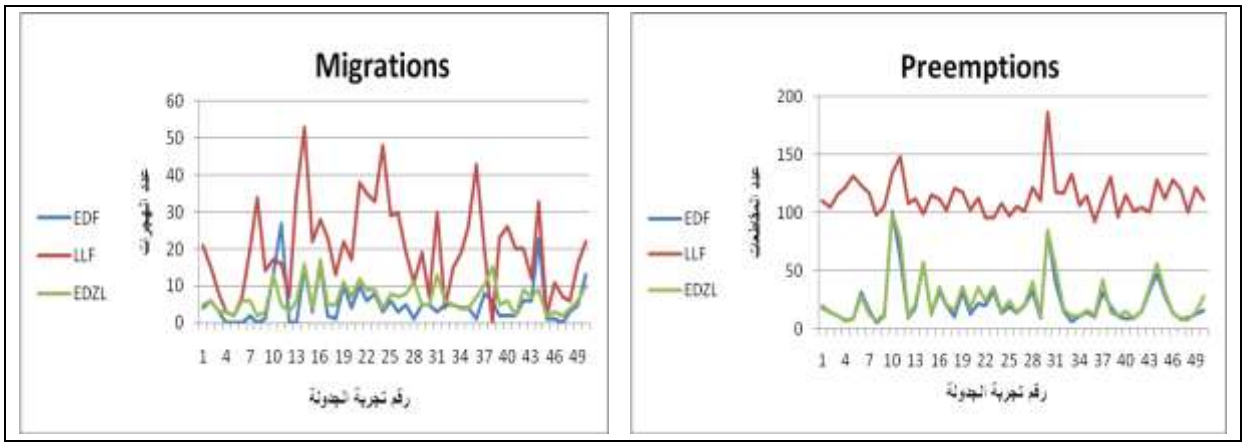


الشكل (5) جدولة أربع مهام دورية على معالجين باستخدام خوارزمية الجدولة EDZL

نلاحظ من الشكل (5) أنه في اللحظة 33 أنه بدأت المهمة Task1 ذات القيد الزمني الدوري الثاني 60 بتنفيذ الدور الثاني لها على المعالج الأول ولكن في اللحظة 35 وردت المهمة Task4 ذات القيد الزمني 42 الأقل من Task1 وبالتالي أصبحت ذات أولوية أعلى وبالتالي قاطعت عمل Task1 وحجزت المعالج. وفي اللحظة 34 أثناء عمل Task4 نلاحظ أن الزمن المتبقي من Task1 لتحقيق الزمن الحدي قد أصبح أقل من الصفر ولذلك عادت Task1 لتقاطع عمل Task4 لتستمر بالعمل على المعالج الأول وبالرغم من ذلك تجاوزت القيد الزمني الحدي 60. هذا هو الفرق بين هذه الخوارزمية والخوارزميتين السابقتين حيث تعمل على إعطاء الأولوية الأعلى للمهمة التي تملك الزمن الحدي الأقل ولكن إذا وردت مهمة أخرى لها زمن متبقي صفري عندها تعطى الأولوية الأعلى ، وبالرغم من ذلك قد يؤدي هذا إلى تجاوز بعض القيود الزمنية للمهام.

تم العمل في هذا البحث على تكرار عملية جدولة أربع مهمات دورية على معالجين متطابقين **خمسين مرة** ، في كل مرة يتم تشكيل مجموعة من أربع مهام ويتم تطبيق خوارزميات الجدولة عليها (كما في المثال السابق) وذلك للتمكن من تحليل أداء الخوارزميات فكانت النتائج كما في الشكل (6) حيث يمثل المحور الأفقي رقم تكرار التجربة ويمثل المحور العمودي قيمة المعامل المدروس :





الشكل (6) نتائج عملية جدولة أربع مهمات دورية على معالجين متطابقين خمسين مرة

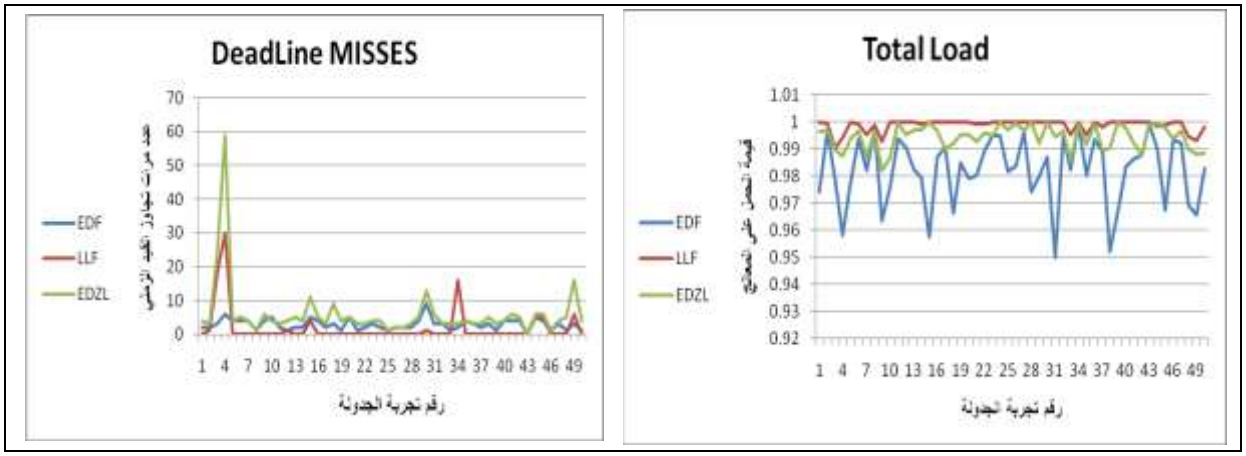
تم حساب مجموع الهجرات ومجموع المقاطعات ومعدل الحمل على المعالج و عدد الإخفاقات في تحقيق القيد الزمني في كل مرة من مرات أداء التجربة ثم تم تقسيم المجموع الناتج على عدد مرات تكرار التجربة (50) للحصول على متوسط حسابي لتحليل ومقارنة أداء الخوارزميات، فكانت النتائج في الجدول (1):

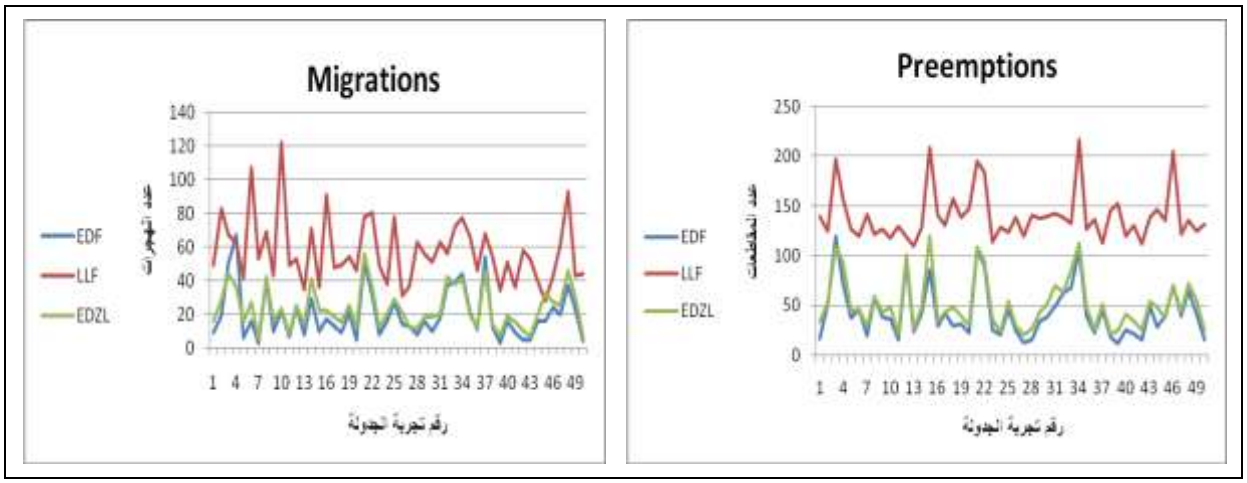
الجدول (1) المتوسط حسابي لقيم معايير أداء الخوارزميات بجدولة أربع مهمات دورية على معالجين متطابقين

4Task_2Processors			
	EDF	LLF	EDZL
Total Load	0.97852	0.997488	0.994568
Preemptions	22.06	113.84	25.24
Migrations	5.18	20.02	6.7
DEAD LINE Misses	2.3	3.24	4.22

4-2 جدولة ثمان مهام دورية (n=8) على أربعة معالجات متطابقة(m=4):

تم في البداية توليد مجموعات مهام عددها مساوي لـ 8 و إنجازية كل مهمة منها أصغر من الواحد و مجموع إنجازيات المهام الأربع مساو لعدد المعالجات أي 4. عمل هذا البحث أيضاً على تكرار عملية جدولة ثمان مهمات دورية على أربعة معالجات متطابقة **خمسين مرة** ، في كل مرة يتم تشكيل مجموعة من ثمان مهام ويتم تطبيق خوارزميات الجدولة عليها (كما في المثال السابق)، وكانت النتائج كما في الشكل (7):





الشكل (7) نتائج عملية جدولة ثمان مهمات دورية على أربعة معالجات متطابقة خمسين مرة

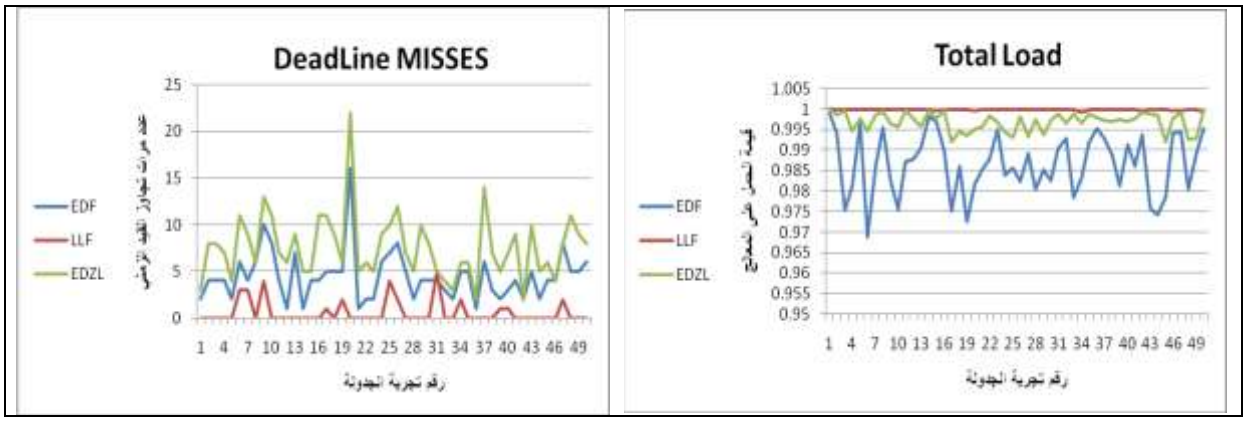
تم حساب مجموع الهجرات ومجموع المقاطعات ومعدل الحمل على المعالج و عدد الإخفاقات في تحقيق القيد الزمني في كل مرة من مرات أداء التجربة ثم تم تقسيم المجموع الناتج على عدد مرات تكرار التجربة (50) للحصول على متوسط حسابي لمقارنة أداء الخوارزميات، فحصلنا على النتائج في الجدول (2):

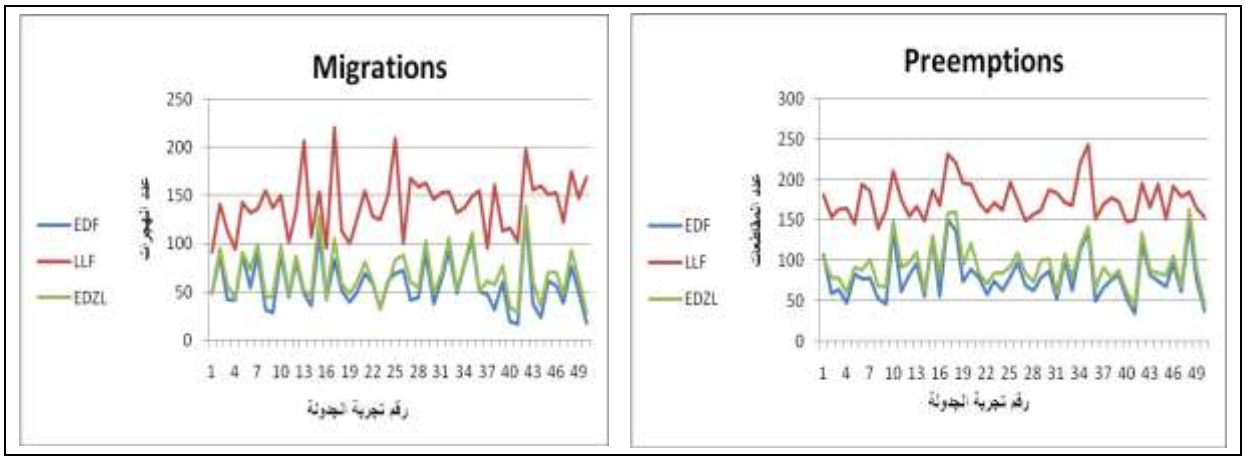
الجدول (2) المتوسط حسابي لقيم معايير أداء الخوارزميات بجدولة 8 مهمات دورية على 4 معالجات متطابقة

8Task_4Processors			
	EDF	LLF	EDZL
Total Load	0.981828	0.998722	0.99425
Preemptions	43.7	140.54	51.12
Migrations	19.88	57.52	23.2
DEAD LINE Misses	2.84	1.74	5.94

4-3 جدولة 16 مهمة دورية (n=16) على ثمان معالجات متطابقة (m=8):

تم في البداية توليد مجموعات مهام عددها مساوي لـ 16 و إنجازية كل مهمة منها أصغر من الواحد و مجموع إنجازيات المهام الأربع مساو لعدد المعالجات أي 8. عمل هذا البحث أيضاً على تكرار عملية جدولة 16 مهمة دورية على ثمان معالجات متطابقة **خمسين مرة** ، في كل مرة يتم تشكيل مجموعة من 16 مهمة ويتم تطبيق خوارزميات الجدولة عليها (كما في الأمثلة السابقة)، وكانت النتائج كما في الشكل (8):





الشكل (8) نتائج عملية جدولة 16 مهمة دورية على 8 معالجات متطابقة خمسين مرة

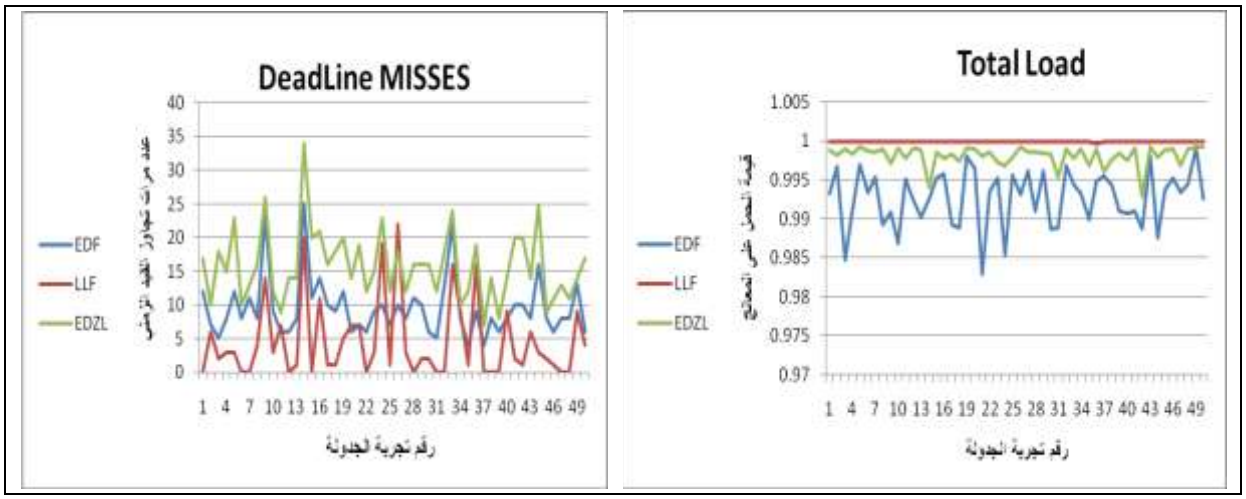
تم حساب مجموع الهجرات ومجموع المقاطعات ومعدل الحمل على المعالج و عدد الإخفاقات في تحقيق القيد الزمني في كل مرة من مرات أداء التجربة ثم تم تقسيم المجموع الناتج على عدد مرات تكرار التجربة (50) للحصول على متوسط حسابي لمقارنة أداء الخوارزميات، حصلنا على النتائج في الجدول (3):

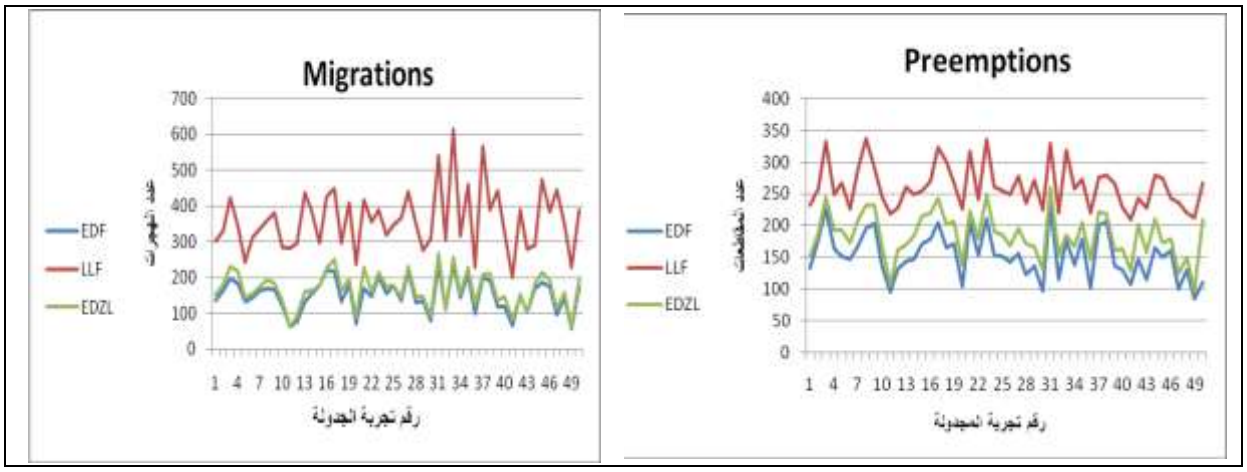
الجدول (3) المتوسط حسابي لقيم معايير أداء الخوارزميات بجدولة 16 مهمة دورية على 8 معالجات متطابقة

16Task_8Processors			
	EDF	LLF	EDZL
Total Load	0.986552	0.999854	0.996784
Preemptions	80.78	175.68	93.52
Migrations	57.72	141.02	67.24
DEAD LINE Misses	4.44	0.6	7.58

4-4 جدولة 32 مهمة دورية (n=32) على 16 معالج متطابق (m=16):

تم في البداية توليد مجموعات مهام عددها مساوي لـ 32 و إنجازية كل مهمة منها أصغر من الواحد و مجموع إنجازيات المهام الأربع مساو لعدد المعالجات أي 16. عمل هذا البحث أيضاً على تكرار عملية جدولة 32 مهمة دورية على 16 معالج متطابق **خمسين مرة** ، في كل مرة يتم تشكيل مجموعة من 32 مهمة ويتم تطبيق خوارزميات الجدولة عليها (كما في الأمثلة السابقة)، وكانت النتائج كما في الشكل (9):





الشكل (9) نتائج عملية جدولة 32 مهمة دورية على 16 معالج متطابق خمسين مرة

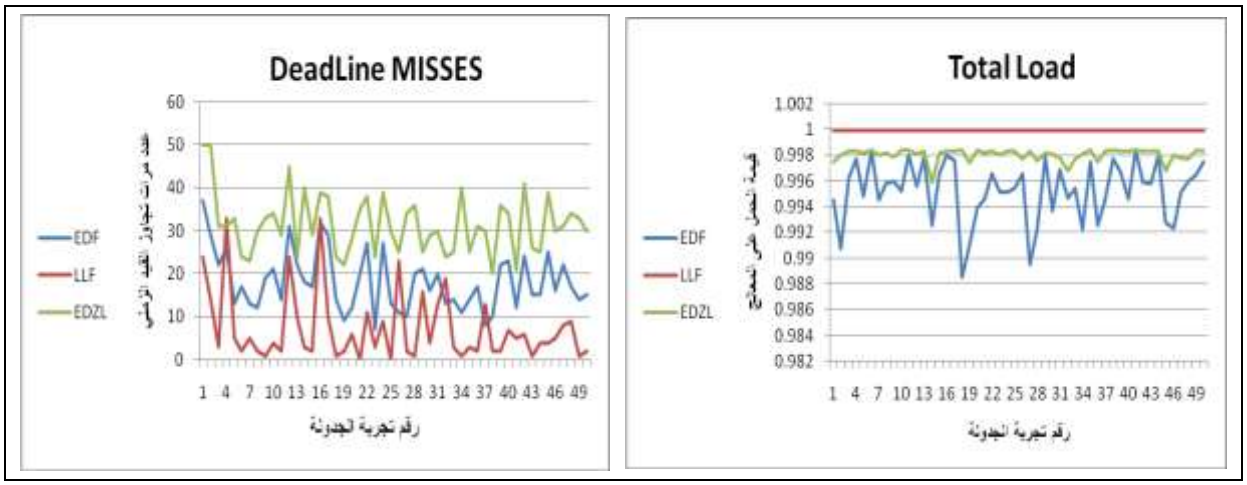
تم حساب مجموع الهجرات ومجموع المقاطعات ومعدل الحمل على المعالج و عدد الإخفاقات في تحقيق القيد الزمني في كل مرة من مرات أداء التجربة ثم تم تقسيم المجموع الناتج على عدد مرات تكرار التجربة (50) للحصول على متوسط حسابي لمقارنة أداء الخوارزميات، حصلنا على النتائج الجدول (4):

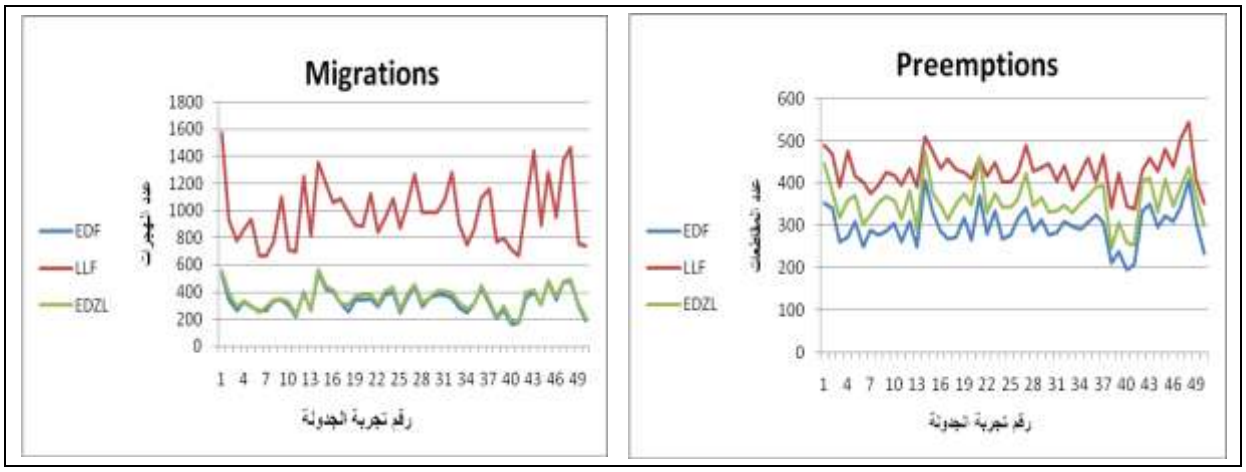
الجدول (4) المتوسط حسابي لقيم معايير أداء الخوارزميات بجدولة 32 مهمة دورية على 16 معالج متطابق

32Task_16Processors			
	EDF	LLF	EDZL
Total Load	0.992696	0.999896	0.998092
Preemptions	152.86	261.94	183.78
Migrations	150.56	359.86	168.2
DEAD LINE Misses	9.48	4.52	15.82

4-5 جدولة 64 مهمة دورية (n=64) على 32 معالج متطابق (m=32):

تم في البداية توليد مجموعات مهام عددها مساوي لـ 64 و إنجازية كل مهمة منها أصغر من الواحد و مجموع إنجازيات المهام الأربع مساو لعدد المعالجات أي 32. عمل هذا البحث أيضاً على تكرار عملية جدولة 64 مهمة دورية على 32 معالج متطابق **خمسين مرة** ، في كل مرة يتم تشكيل مجموعة من 64 مهمة ويتم تطبيق خوارزميات الجدولة عليها (كما في الأمثلة السابقة)، وكانت النتائج كما في الشكل (10):





الشكل (10) نتائج عملية جدولة 64 مهمة دورية على 32 معالج متطابق خمسين مرة

تم حساب مجموع الهجرات ومجموع المقاطعات ومعدل الحمل على المعالج و عدد الإخفاقات في تحقيق القيد الزمني في كل مرة من مرات أداء التجربة ثم تم تقسيم المجموع الناتج على عدد مرات تكرار التجربة (50) للحصول على متوسط حسابي لمقارنة أداء الخوارزميات، حصلنا على النتائج في الجدول(5):

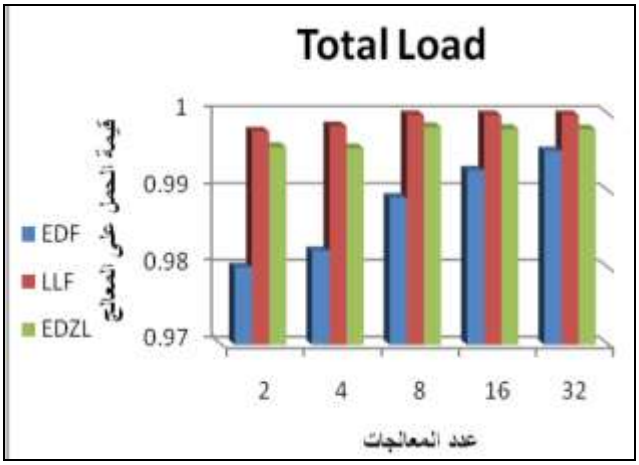
الجدول (5) المتوسط حسابي لقيم معايير أداء الخوارزميات بجدولة 64 مهمة دورية على 32 معالج متطابق

64Task_32Processors			
	EDF	LLF	EDZL
Total Load	0.99522	0.9999	0.998036
Preemptions	297.08	429.62	356.68
Migrations	331.72	987.66	353.48
DEAD LINE Misses	18.12	7.3	31.54

6-4 ملخص النتائج:

يمثل الشكل (11) معدل الحمل على المعالجات لخوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات المدروسة:

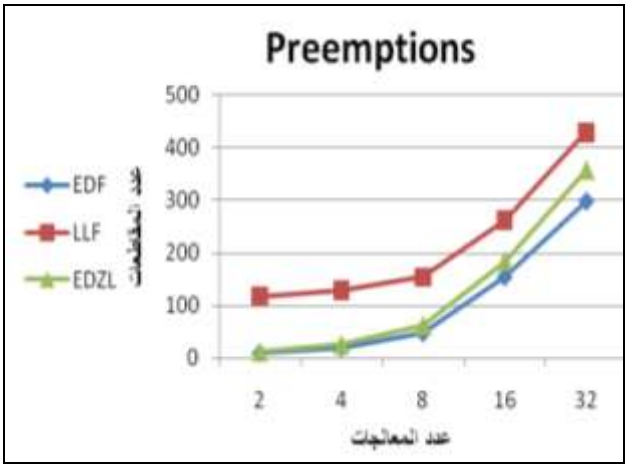
Total Load			
No.Processors	EDF	LLF	EDZL
2	0.97852	0.997488	0.994568
4	0.981828	0.998722	0.99425
8	0.986552	0.999854	0.996784
16	0.992696	0.999896	0.998092
32	0.99522	0.9999	0.998036



الشكل (11) معدل الحمل على المعالجات لخوارزميات الجدولة بالنسبة لكافة المعالجات

يظهر الشكل (12) عدد المقاطعات الناتجة عن خوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات المدروسة:

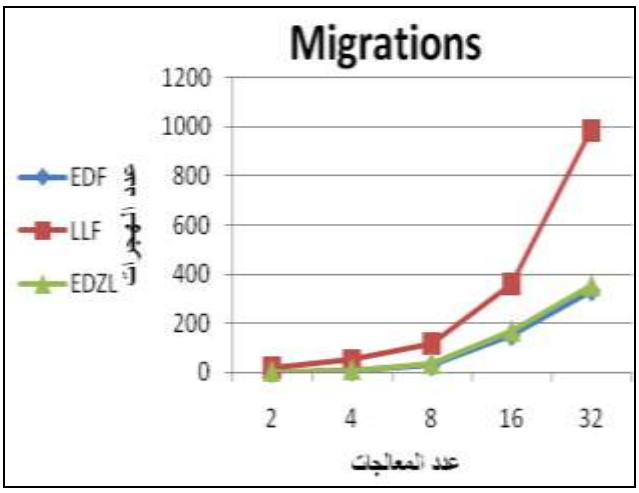
Preemptions			
No.Processors	EDF	LLF	EDZL
2	22.06	113.84	25.24
4	43.7	140.54	51.12
8	80.78	175.68	93.52
16	152.86	261.94	183.78
32	297.08	429.62	356.68



الشكل (12) عدد المقاطعات الناتجة عن خوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات

يبين الشكل (13) عدد الهجرات الناتجة عن خوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات المدروسة.

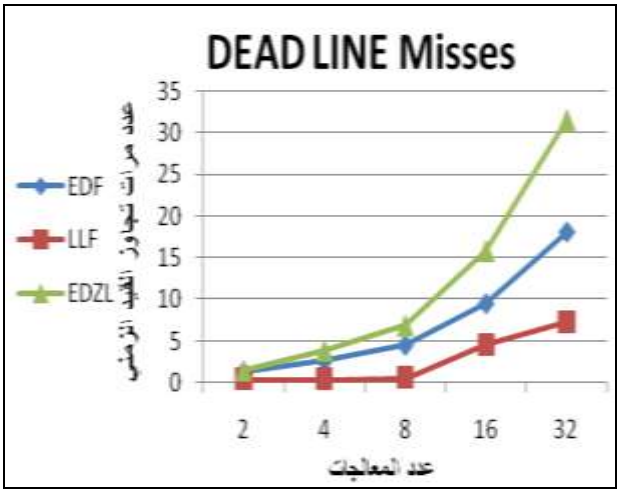
Migrations			
No.Processors	EDF	LLF	EDZL
2	5.18	20.02	6.7
4	19.88	57.52	23.2
8	57.72	141.02	67.24
16	150.56	359.86	168.2
32	331.72	987.66	353.48



الشكل (13) عدد الهجرات الناتجة عن خوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات

يظهر الشكل(14) عدد الإخفاقات في تحقيق القيود الزمنية لخوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات المدروسة.

DEAD LINE Misses			
No.Processors	EDF	LLF	EDZL
2	2.3	3.24	4.22
4	2.84	1.74	5.94
8	4.44	0.6	7.58
16	9.48	4.52	15.82
32	18.12	7.3	31.54



الشكل (14) عدد الإخفاقات في تحقيق القيود الزمنية الناتجة عن خوارزميات الجدولة الثلاث بالنسبة لكافة المعالجات

4-7 مقارنة النمذجة و تحليل الأداء في هذا البحث مع دراسات سابقة:

قدمت الكثير من الدراسات السابقة تحليلاً لأداء العديد من خوارزميات الجدولة في أنظمة الزمن الحقيقي متعددة المعالجات ، وعلى سبيل المثال لا الحصر نوضح في الجدول (6) مقارنة بين عدة دراسات من حيث أنواع خوارزميات المقارنة، وأنواع المهام المستخدمة في الجدولة و المعايير التي تم على أساسها قياس الأداء للخوارزميات المدروسة:

الجدول (6) مقارنة النمذجة وتحليل الأداء في هذا البحث مع دراسات سابقة

	الخوارزميات المدروسة			المهام		معايير الأداء التي تمت المقارنة بها				
	EDF	LLF	EDZL	عددها	نوعها	الحمل على المعالجات	المقاطعات	الهجرات	القيد الزمني للمهام	معايير أخرى
هذا البحث	✓	✓	✓	4-64	دورية	✓	✓	✓	✓	
[9]	✓		✓	24	دورية		✓			✓
[11]	✓	✓	✓	4500	شبه دورية	✓				✓
[12]	✓	✓		100	دورية	✓				✓

يبين الجدول (6) المقارنة بين هذا البحث والدراسات السابقة ، فالدراسة المرجعية [9] عملت على المقارنة بين الخوارزميتين EDF و EDZL على 24 مهمة دورية من حيث عدد المقاطعات وعدد المهام التي دخلت بعملية الجدولة،

حيث أظهرت الدراسة بأن عدد المقاطعات متقارب جداً بين الخوارزميتين وأن الخوارزمية EDZL كانت أفضل من ناحية جدولة عدد مهام أكبر من EDF.

أما الدراسة المرجعية [11] عملت على المقارنة بين الخوارزميتين EDF و EDZL و LLF على عدد كبير من المهام وصل إلى 4500 مهمة شبه دورية، حيث درست المقارنة بين الخوارزميات من حيث عدد المهام التي دخلت بعملية الجدولة ومن ناحية الحمل على المعالجات حيث بينت بأن LLF هي استطاعت جدولت أكبر عدد من المهام وأن EDF كانت الأقل حملاً على المعالجات.

أما الدراسة المرجعية [12] عملت على المقارنة بين الخوارزميتين LLF و EDF بالنسبة لعدد المهام التي تمت جدولتها بنجاح وذلك على عدد متزايد من المهام وصل إلى 100 مهمة دورية ، حيث درست تأثير الحمل على المعالجات وتأثير زيادة عدد المعالجات والمهام المتزايد على نسبة نجاح الجدولة.

قدم هذا البحث نمذجة وتحليل أداء بشكل مفصل لأداء الخوارزميات LLF و EDZL و EDF لجدولة المهام الدورية بالنسبة لعدة معايير أداء هامة بشكل أكبر وأوسع من دراسات عديدة قدمت سابقاً في هذا المجال.

الاستنتاجات و التوصيات :

نستنتج من ملخص النتائج بأن خوارزمية LLF هي الأفضل من ناحية الحمل على المعالجات وهذا الحمل يزداد بشكل طفيف بازدياد عدد المهام والمعالجات، أي تستغل هذه الخوارزمية وقت المعالجات بشكل كبير، في الوقت الذي نجد فيه أنها تتسبب في عدد كبير جداً من المقاطعات والهجرات للمهام وبالتالي تزيد من كلفة ومن طول الجدولة، أما من ناحية تحقيق القيود الزمنية للمهام الدورية نجد بأنها الأفضل لأنها الأقل إخفاقاُ في تحقيق هذه القيود.

أما بالنسبة للخوارزمية الأقل قدرة على استغلال وقت المعالجات أثناء الجدولة فهي EDF ولكنه يتزايد بشكل كبير بازدياد عدد المهام والمعالجات، ونجد أنها الخوارزمية الأفضل من ناحية عدد المقاطعات وعدد الإخفاقات في تحقيق القيود الزمنية للمهام من الخوارزمية EDZL وتتقارب جداً معها بعدد الهجرات.

تبقى الخوارزمية EDZL متوسطة الحمل على المعالجات مقارنة بالخوارزميتين السابقتين ولكنها الأكثر إخفاقاً في تحقيق القيود الزمنية للمهام.

نجد بأن تطبيق خوارزمية LLF في منظومات الزمن الحقيقي الصلبة أكثر أمناً حيث تظهر النتائج بأن عدد القيود الزمنية التي لم تستطع هذه الخوارزمية تحقيقها أقل بكثير من الخوارزميتين EDF و EDZL ، ويمكن تطبيق الخوارزميتين EDF و EDZL في منظومات الزمن الحقيقي اللينة التي لا تؤدي إلى نتائج كارثية في حال الإخفاق بتحقيق القيود الزمنية للمهام.

تفضل الخوارزمية EDF في حال كانت كلفة الجدولة وطولها أكثر أهمية من تحقيقها لعدد كبير من القيود الزمنية ويبين الجدول (7) الخوارزمية الأفضل بالنسبة لكل معيار من معايير الأداء.

الجدول (7) يبين الخوارزمية الأفضل بالنسبة لكل معيار من معايير الأداء

	EDF	LLF	EDZL
Total Load		✓	
Preemptions	✓		
Migrations	✓		✓
DEAD LINE Misses		✓	

التوصيات:

إن عملية الجدولة للمهام الدورية على عدة معالجات في نظم الزمن الحقيقي ليست عملية سهلة، كما أن مقدار العمل البحثي المخصص للتقنيات المُصمَّمة لتحليل وتطوير الخوارزميات المتوفرة يشهد على صحة ذلك.

يقترح للدراسات المستقبلية:

1 -تحليل أداء خوارزميات أخرى ومقارنة النتائج مع الخوارزميات المدروسة.

2 -جدولة أنواع أخرى من المهام مثل المهام اللادورية وشبه الدورية.

3 -تغيير قيمة الإنجازية الكلية لكافة المهام ($U_{sum}(T) <= m$) لتصبح أقل من الحد الأقصى المسموح به الذي اعتمده البحث وهو $U_{sum}(T) = m$.

4 -تغيير نسبة عدد المعالجات m إلى عدد المهام n لتصبح أقل من ($n=2m$) لإيجاد عدد المعالجات الأصغري الذي تبقى عنده جميع المهام قابلة للجدولة.

المراجع:

[1] M. Joseph, *"Real-time Systems: Specification, Verification and Analysis*" , Prentice Hall,1996.

[2] J. A. Stankovic and K. Ramamritham, *"Tutorial Hard Real-Time Systems"*, IEEE Computer Society Press, 1998.

[3] M .Arezou and A. G.Selim, *"Scheduling Algorithms for Real-Time Systems"*, School of Computing , Canada, 2005.

[4] G. C. Buttazzo, *"Hard Real-Time Computing Systems: predictable scheduling algorithmsand applications"*, Springer company, 2005.

[5] Alhussian.H, Zakari.N, Hussin.F.A ,*"An Efficient Real-Time Multiprocessor Scheduling Algorithm"*, Journal of Convergence Information Technology (JCIT) Volume9, Number2, March 2014.

[6] Bertogna.M, Cirinei.M, Lipari.G ,"*Schedulability analysis of global scheduling algorithms on multiprocessor platforms*", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. X, NO. X, JUNE 2008.

[7] J. A. Stankovic, M. Spuri, K. Ramamritham, and G. C. Buttazzo, *"Deadline Scheduling for Real-Time Systems, EDF and related algorithms*", Kluwer Academia Publishers, 1998.

[8] Behera.S , Khuntia.S ,Nayak.S , *" An Improved Least-Laxity-First Scheduling Algorithm For Real-Time Tasks* ",International Journal of Engineering Science and Technology (IJEST),April ,2012.

[9] PARK.M, HAN.S ,KIM.H,*" Comparison of Deadline-Based Scheduling Algorithms for Periodic Real-Time Tasks on Multiprocessor"*, IEICE TRANS. INF. & SYST., VOL.E88–D, NO.3 MARCH 2005.

[10] Cheramy.M , Deplanche .A and Hladik.P ,*"Simulation of Real-Time Multiprocessor Scheduling with Overheads* " , published in SIMULTECH, Reykjavik : Islande ,2013.

[11] Lee.J , Easwaran.A , Shin.I ,*" Laxity dynamics and LLF schedulability analysis on multiprocessor platforms "*, Springer Science Business Media, LLC 2012.

[12] Salmani.V , Naghibzadeh.M ,Habibi.A , Deldari.H," *Quantitative Comparison of Job-level Dynamic Scheduling Policies in Parallel Real-time Systems"* ,Department of Computer Engineering, Ferdowsi University of Mashhad, Iran,2004.