الجمهورية العربية السورية

جـــــامعة دمـــشـــق

كلية الهندسة الميكانيكية و الكهربائية

قسم هندسة الحواسيب والأتمتة

تحسين خوارزمية جدولة التغذية العكسية متعددة المستويات (MLFQ) في أنظمة الزمن الحقيقي

رسالة أُعدت لنيل درجة الماجستير في هندسة الحواسيب وشبكاتها

إعداد

المهندس باسل علي أسعد

بإشراف

الدكتورة رأفة خازم

أستاذ في قسم هندسة الحواسيب والأتمتة

2024-2023

الفهرس

1 الفصل الأول: المقدمة 10

1-1 مقدمة 11

1-1-1 جدولة الإجرائيات 12

1-1-2 جدولة المعالج (الجدولة قصيرة الأمد) 13

1-1-3 حالة الإجرائية Process State 13

1-1-4 معايير الجدولة 15

1-1-5 أهداف خوارزميات الجدولة 16

1-2 أهمية البحث 20

1-3 الهدف من البحث 21

1-4 المسألة التي يناقشها البحث 21

1-5 الدراسات المرجعية 21

2 الفصل الثاني: الدراسة النظرية 25

2-1 خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ 26

2-1-1 آلية عمل خوارزمية MLFQ 27

2-1-2 معايير تقييم الأداء وعرض النتائج 36

2-1-3 مثال 38

2-2 خوارزمية جدولة التغذية العكسية متعددة المستويات المُحسنة سابقاً MLFQ-RT 42

2-2-1 آلية عمل خوارزمية MLFQ-RT 45

2-2-2 معايير تقييم الأداء وعرض النتائج 50
2-2-3 مثال 51
2-3 خوارزمية الزمن النهائي الأقرب أولاً Earliest Deadline First (EDF) 55
2-4 الخوارزمية المُقترحة MLFQ-EDF 56
2-4-1 آلية عمل الخوارزمية المقترحة MLFQ-EDF 57
2-4-2 معايير تقييم الأداء وعرض النتائج 67
3 الفصل الثالث: التطبيق العملي 69
3-1 بيئة التطوير وتنجيز الخوارزميات 70
3-1-1 تنجيز خوارزمية MLFQ 79
3-1-2 تنجيز خوارزمية MLFQ-RT 80
3-1-3 تنجيز الخوارزمية المُقترحة MLFQ-EDF 83
3-2 النتائج العملية 89
3-2-1 الاختبار الأول (المجموعة الأولى) 89
3-2-2 الاختبار الثاني (المجموعة الثانية) 100
4 النتائج والآفاق المستقبلية 112
5 المراجع 113

قائمة الأشكال

الشكل (1-1): رشقات وحدة المعالجة المركزية ورشقات I/O الخاصة بإجرائية معينة 12

الشكل (1-2): الحالات التي تمر بها الإجرائية.....15

الشكل (1-3): مستويات رتل الجهوزية.....18

الشكل (2-1): خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ 27

الشكل (2-2): مخطط عمل خوارزمية MLFQ 29

الشكل (2-3): مرحلة اختيار الإجرائية المناسبة واسنادها إلى وحدة المعالجة المركزية 30

الشكل (2-4): مرحلة اختيار الإجرائية الأعلى أولوية Highest Priority 33

الشكل (2-5): مخطط الخوارزمية MLFQ 38

الشكل (2 6): مخطط آلية عمل خوارزمية MLFQ-RT 44

الشكل (2-7): خوارزمية الزمن الحرج الأقرب أولاً EDF 55

الشكل (2-8): مخطط آلية عمل الخوارزمية المُقترحة MLFQ-EDF 57

الشكل (2 9): المخطط التدفقي للخوارزمية المُقترحة MLFQ-EDF 60

الشكل (2-10): مرحلة اختبار خصائص الإجرائية قبل إدخالها إلى أرتال الجهوزية 61

الشكل (3-1): الإجرائيات التي تم التجريب عليها في الاختبار الأول 90

الشكل (3-2): تحسن نسبة نجاح إجرائيات الزمن الحقيقي (الاختبار الأول) MLFQ-EDF 95

الشكل (3-3): تحسن متوسط أزمنة انتظار الإجرائيات (الاختبار الأول) MLFQ-EDF 97

الشكل (3-4): تحسن معيار متوسط تنفيذ الإجرائيات (الاختبار الأول) في خوارزمية MLFQ-EDF.... 99

الشكل (3-5): الإجرائيات التي تم الاختبار عليها في الاختبار الثاني 101

الشكل (3-6): تحسن نسبة نجاح إجرائيات الزمن الحقيقي (الاختبار الثاني) MLFQ-EDF..... 106

الشكل(3-7): تحسين معيار متوسط أزمنة انتظار الإجرائيات (الاختبار الثاني) MLFQ-EDF..... 108

الشكل(3-8): تحسن متوسط أزمنة التنفيذ الكلية للإجرائيات (الاختبار الثاني) MLFQ-EDF 110

قائمة الجداول

الجدول (2–1): خصائص عينات الدخل لخوارزمية MLFQ 40

الجدول (2–2): أزمنة الإجرائيات بعد تنفيذ خوارزمية MLFQ 42

الجدول (2–3): خصائص عينات الدخل لخوارزمية MLFQ-RT 52

الجدول (2–4): أزمنة الإجرائيات بعد تنفيذ خوارزمية MLFQ-RT 53

الجدول (2–5): حالة الإجرائية الحالية وأرتال الجهوزية Ready Queues 66

الجدول (3–1): الصفوف المستخدمة في بيئة التطوير 70

الجدول (3–2): خصائص الإجرائية ضمن الصف Process_Data 71

الجدول (3–3): بنى المعطيات والمتغيرات المستخدمة لتقييم أداء الخوارزمية المُقترحة 72

الجدول (3–4): بنى المعطيات والمتغيرات المستخدمة في تنفيذ الإجرائيات وبناء المُجدولات 73

الجدول (3–5): التوابع المُستخدمة لترتيب الإجرائيات ضمن أرتال الجهوزية Ready Queues 74

الجدول (3–6): خصائص الإجرائيات Processes 76

الجدول (3–7): التوابع المُمثلة لكل من خوارزميات الجدولة الثلاثة 78

الجدول (3–8): أرتال الجهوزية المستخدمة لخوارزمية MLFQ-RT 81

الجدول (3–9): بنى المعطيات الممثلة لأرتال الجهوزية Ready queues 84

الجدول (3–10): نتائج الاختبار الأول على خوارزمية MLFQ 91

الجدول (3–11): نتائج الاختبار الأول على خوارزمية MLFQ-RT 92

الجدول (3–12): نتائج الاختبار الأول على الخوارزمية المقترحة MLFQ-EDF 93

الجدول (3–13): مقارنة نسبة النجاح Success Ratio (الاختبار الأول) بين الخوارزميات الثلاثة.... 94

الجدول (3–14): مقارنة نسب تحسين متوسط أزمنة انتظار الإجرائيات (الاختبار الأول) 96

الجدول (3–15): مقارنة نسب تحسين متوسط أزمنة التنفيذ الكلية للإجرائيات (الاختبار الأول). 98

الجدول (3–16): نسب التحسين في الاختبار الأول 99

الجدول (3-17): نتائج الاختبار الثاني على خوارزمية MLFQ 102
الجدول (3-19): نتائج الاختبار الثاني على الخوارزمية المقترحة MLFQ-EDF 104
الجدول (3-20): مقارنة نسبة نجاح إجرائيات الزمن الحقيقي (الاختبار الثاني). 105
الجدول (3-21): مقارنة نسب تحسن متوسط أزمنة انتظار الإجرائيات (الاختبار الثاني) 107
الجدول (3-22): مقارنة نسب تحسن متوسط أزمنة التنفيذ الكلية للإجرائيات (الاختبار الثاني)..... 109
الجدول (3-23): نسب التحسين في الاختبار الثاني 110

جدول الاختصارات

الاختصار	معنى الاختصار	شرح
FIFO	First In First Out	نمط الداخل أولاً يخرج أولاً
RTS	Real Time Systems	أنظمة الزمن الحقيقي
VM	Virtual Machine	الجهاز أو الآلة الافتراضية
WT	Waiting Time	زمن الانتظار
TT	Turnaround Time	زمن التنفيذ (زمن الإقامة)
RAM	Random access memory	ذاكرة الوصول العشوائي
TET	Total Execution Time	زمن التنفيذ الكلي
SRTF	Shortest Remainig Time First	خوارزمية الزمن المتبقي الأقصر أولاً
R-time	Process Remaining Time	زمن الإجرائية المتبقي
wt	Process Waiting Time	زمن الانتظار
PCB	Process Control Block	كتلة التحكم بالإجرائية
P_type	Process_Type	نوع الإجرائية (زمن حقيقي أو عادية)
CLK	CPU Clock	دورة الساعة الحالية لـ CPU

ملخص

تُعد جدولة المهام أمراً بالغ الأهمية في الأنظمة الحاسوبية، واختيار خوارزمية جدولة المهام المناسبة لضمان تنفيذ الإجرائيات بأقل زمن انتظار وأصغر زمن تنفيذ موضوعاً هاماً جداً، وفي أنظمة الزمن الحقيقي Real Time Systems يكون التركيز على ضمان تنفيذ الإجرائيات ضمن الحدود الزمنية الحرجة Deadlines من خلال جدولة جميع المهام بمعدل نجاح أعظمي لإجرائيات الزمن الحقيقي Success Ratio مقارنتها ببقية الإجرائيات التي ليست في الزمن الحقيقي.

يقترح هذا البحث خوارزمية الجدولة MLFQ-EDF كتحسين لخوارزمية من خوارزميات جدولة وحدة المعالجة المركزية CPU هي خوارزمية الرتل متعدد المستويات وذات التغذية العكسية MLFQ (Multilevel Feedback Queue) وذلك لتحسين أداء هذه الخوارزمية من خلال زيادة نسبة النجاح Success Ratio وتقليل متوسط زمن الانتظار ومتوسط زمن التنفيذ الكلي.

تدمج الخوارزمية المقترحة بين خوارزميتي جدولة هما خوارزمية الزمن الحرج الأقرب أولاً EDF (Earliest Deadline First) وخوارزمية MLFQ حيث تُطبق خوارزمية EDF في رتل الجهوزية الأعلى مستوى من خوارزمية MLFQ بهدف الحصول على أفضل نسبة نجاح لإجرائيات الزمن الحقيقي وأقل متوسط لزمن الانتظار وزمن التنفيذ الكلي وذلك من خلال مقارنة تنجيز وتجريب الخوارزمية المقترحة MLFQ-EDF مع خوارزمية MLFQ والنسخة المُحسنة سابقاً عنها MLFQ-RT.

تم تنجيز كل خوارزمية من الخوارزميات ومحاكاتها باستخدام باستخدام لغة C++ وبيئة Visual Studio.

بينتْ النتائج – بعد تنجيز الخوارزمية المقترحة MLFQ-EDF وتجريبها– تحسن معدل نجاح إجرائيات الزمن الحقيقي بنسبة 25% وتحسين معدل زمن الانتظار بنسبة 23% وتحسين معدل زمن التنفيذ بنسبة 24% مقارنة بخوارزمية MLFQ، ومعدل نجاح إجرائيات الزمن الحقيقي بنسبة 17.3% وتحسين معدل زمن الانتظار بنسبة 8.72% وتحسين معدل زمن التنفيذ بنسبة 8% مقارنة بخوارزمية MLFQ-RT المُحَسنة سابقاً.

الفصل الأول: المقدمة

1-1 مقدمة

جدولة المعالج هي إحدى السياسات المعتمدة في نظام التشغيل لضمان عدم انتظار أية إجرائية من إجرائيات الحاسب لأجل غير مسمى لاستحواذها على المعالج وضمان العدل في اقتسام زمن المعالج (وحدة المعالجة المركزية CPU) بين الإجرائيات أي أن جدولة المعالج مطلب أساسي لزيادة مدى الاستفادة من المعالج إلى الحد الأقصى الأمر الذي يؤثر بشكل مباشر على أداء النظام واستخدام الموارد بالشكل الأمثل والأكثر فاعلية.

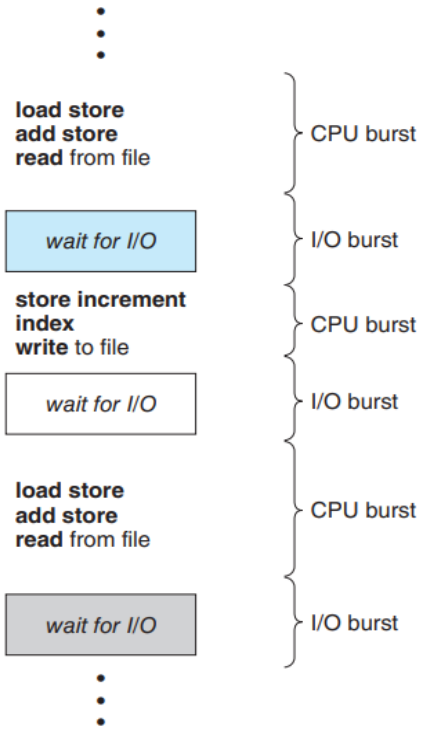
ونظراً لكون الإجرائيات الحاسوبية أو المهام تتنافس على الاستفادة من وحدة المعالجة المركزية أدى هذا إلى ظهور الحاجة إلى خوارزميات الجدولة لإسناد الإجرائيات بشكل عادل لوحدات المعالجة المركزية المتاحة بهدف تحسين مقاييس الأداء على وجه الخصوص وزيادة الاستخدام الإجمالي والإنتاجية وتقليل وقت الاستجابة ووقت الانتظار [4].

تعتبر الجدولة واحدة من المهام الأساسية لنظام التشغيل حتى يمكننا القول تقريباً أغلب موارد النظام يتم جدولتها قبل الاستخدام [19]. ومن الجوانب الأكثر أهمية في التقنيات الحاسوبية الحديثة هي قدرتها على تنفيذ العديد من الأنشطة والمهام في وقت واحد من خلال استخدام خوارزميات الجدولة بالشكل الفعال [8].

وبمجرد أن تصبح حالة وحدة المعالجة المركزية خاملة Idle، يقوم نظام التشغيل باختيار إحدى المهام (الإجرائيات) من رتل الجهوزية Ready queue ليتم تنفيذها على المعالج، وقد تختلف بنية رتل الجهوزية فقد يكون FIFO أو بنية شجرية أو بنية يتم الاختيار منها وفق الأولويات Priorities وهذا مرتبط بخوارزمية الجدولة التي يعتمدها نظام التشغيل لجدولة المعالج.

يتم تنجيز رتل الجهوزية غالباً كقائمة مترابطة من العقد التي تمثل كل منها كتلة التحكم بالإجرائية PCB وبعد جدولة الإجرائية وخلال تنفيذها على المعالج، تمر الإجرائيات بفترات تكون فيها إما تُنفذ على المعالج (رشقات المعالج) أو تقوم بعملية إدخال أو إخراج (رشقة I/O).

ويوضح الشكل (1-1) رشقات وحدة المعالجة المركزية ورشقات I/O الخاصة بإجرائية معينة.



الشكل (1-1): رشقات وحدة المعالجة المركزية ورشقات I/O الخاصة بإجرائية معينة.

1-1-1 جدولة الإجرائيات

ويمكن تصنيف الجدولة في نُظم التشغيل إلى أربعة أنواع [20]:

- الجدولة طويلة الأمد (جدول الأعمال) Long-term (Job) scheduling.

وهي الجدولة التي تنطوي على اتخاذ القرار بإضافة إجرائية إلى مجموعة الإجرائيات التي تم الاعتراف بها في النظام وبمجرد الاعتراف بالإجرائية تتم إضافتها إلى رتل الجهوزية Ready queue (مجموعة الإجرائيات المحملة في الذاكرة وتنتظر تنفيذها على المعالج)، أي في هذه الجدولة يتم نقل الإجرائيات من القرص الصلب إلى الذاكرة.

- الجدولة متوسطة الأمد Medium-term scheduling .

مسؤولة عن اتخاذ القرار بإضافة عدد من الإجرائيات جزئياً أو كلياً في الذاكرة الرئيسية أي تتعلق بالإجرائيات المشاركة في عملية المبادلة Swapping.

- الجدولة قصيرة الأمد Short-term scheduling.

وهي الجدولة التي تنطوي على اتخاذ القرار باختيار الإجرائية التي ستنفذ على المعالج من بين الإجرائيات في رتل الجهوزية Ready queue.

- جدولة I/O scheduling.

وهي الجدولة المسؤولة عن التعامل مع طلبات الإدخال والإخراج على الإجرائية.

1-1-2 جدولة المعالج (الجدولة قصيرة الأمد)

وهناك أربعة ظروف تتطلب من نظام التشغيل اتخاذ قرار باختيار الإجرائية من رتل الجهوزية لتتم جدولتها على المعالج [19]:

1. عندما تنتقل الإجرائية من حالة التنفيذ Running إلى حالة الانتظار Waiting.
2. عندما تنتقل الإجرائية من حالة التشغيل Running إلى حالة الجهوزية Ready.
3. عندما تنتقل الإجرائية من حالة الانتظار Waiting إلى حالة الجهوزية Ready.
4. عندما تنتهي الإجرائية.

في حال كانت الظروف التي تستدعي اتخاذ القرار من قبل نظام التشغيل هي 1 و 4 فقط، نقول عن الجدولة أنها غير شفعية non-preemptive أي بمجرد استحواذ الإجرائية على المعالج لن تتركه إلا بشكل طوعي (انتهاء التنفيذ – طلب I/O) ولا تتم مقاطعتها أبداً. وفي حال كانت الظروف هي 2 و3 بالإضافة إلى 1 و 4، نقول عن الجدولة أنها شفعية (قابة للاستيلاء) preemptive أي يمكن مقاطعة الإجرائية خلال تنفيذها.

1-1-3 حالة الإجرائية Process State

تُعرف حالة الإجرائية بأنها النشاط الحالي الذي تقوم به الإجرائية. وخلال دورة حياتها تمر بمجموعة حالات. وسنذكر هنا الحالات الأكثر شيوعاً وهي:

- جديدة New.

وهي حالة الإجرائية عند اعتراف نظام التشغيل بها. وهي إجرائية تدخل إلى النظام ولكنها غير جاهزة بعد للتنفيذ، ويقوم نظام التشغيل بتهيئة الإجرائية ونقلها إلى حالة الجهوزية.

- جاهزة Ready.

وهي حالة إجرائية تم تحميلها في الذاكرة وتنتظر تسليمها إلى المعالج.

- قيد التشغيل Running.

وهي إجرائية سُلمت إلى المعالج ويتم تنفيذ تعليماتها.

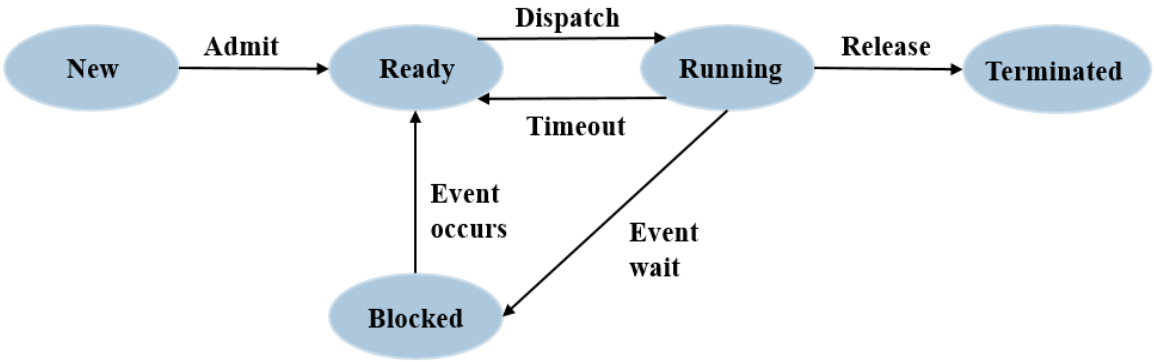
- منتظرة Waiting.

وهي حالة إجرائية تم تعليق تنفيذها بانتظار وقوع حدث ما (تحرير مورد ما).

- منتهية Terminated.

وهي حالة إجرائية انتهى تنفيذها بشكل طبيعي أو غير طبيعي.

ويوضح الشكل (1-2) الحالات التي تمر بها الإجرائية:



الشكل (1-2): الحالات التي تمر بها الإجرائية

1-1-4 معايير الجدولة

وهي المعايير التي أُقترحت لتساعد المطوريين على اختبار خوارزمية الجدولة [19]، التي سيتم اعتمادها في نظام التشغيل قيد التطوير. وهذه المعايير تَشمل:

يُوجد العديد من المعايير لمقارنة خورازميات جدولة وحدة المعالجة المركزية [19]. وبناءاً على هذه المعايير يمكننا تحديد فيما إذا كانت هذه الخوارزمية المعينة أفضل من خوارزمية أخرى أم لا، تتضمن المعايير مايلي:

- مدى الاستفادة من وحدة المعالجة المركزية CPU utilization.

وهي النسبة المئوية للفترة الزمنية التي فيها المعالج مشغولاً خلال فترة تشغيله. وهو معيار يشير إلى مدى انشغالية المعالج وزيادتُه تُحسن من أداء النظام. نظرياً تتراوح انشغالية المعالج من 0 إلى 100 أما عملياً فتتراوح من 40 إلى 90 (نظام محمل بشكل زائد).

- المردود Throughput.

وهو عدد الإجرائيات التي انتهى تنفيذها خلال واحدة الزمن.

- زمن التنفيذ الكلي Turnaround time.

وهو المدة التي تستغرقها الإجرائية في التنفيذ من وقت تسليمها إلى انتهائها. ويُعرف أيضاً بأنه مجموع الفترات التي تقضيها الإجرائية وهي تنتظر في رتل الجاهزية Ready Queue مضافاً لها وقت تنفيذها على المعالج ووقت القيام بعمليات الإدخال والإخراج I/O.

- زمن الانتظار Waiting time.

وهو الزمن الذي تقضيه الإجرائية وهي تنتظر في رتل الجهوزية Ready queue.

- زمن الاستجابة Response time.

وهو الفترة الزمنية التي تستغرقها الإجرائية للبدء بالتنفيذ والاستجابة لها وتسعى خوارزميات الجدولة إلى تقليل زمن الاستجابة قدر الإمكان.

- تبديل السياق Context switch.

وهو العملية التي يقوم بها نظام التشغيل (مسند المهام) لتبديل المعالج بين إجرائيتين وتنطوي هذه العملية على عمليتين: تخزين حالة السياق للإجرائية التي كانت تنفذ على المعالج State save، وتحميل حالة السياق للإجرائية التي ستنفذ على المعالج State restore، وحالة السياق للإجرائية يتم تحميلها في كتلة التحكم بالإجرائية (PCB) [9].

1-1-5 أهداف خوارزميات الجدولة

من أجل تصميم واختيار خوارزمية الجدولة من الضروري وجود هدف محدد من خوارزمية الجدولة المُختارة فقد يكون النظام دفعي Batch أو نظام تفاعلي Interactive أو نظام زمن حقيقي Real time [21].

الجدولة في الأنظمة الدفعية Batch systems.

في هذا القسم سوف نلقي نظرة على أبرز خورازميات الجدولة المستخدمة في الأنظمة الدفعية.

- خوارزمية الجدولة من يأتي أولاً، يُخدم أولاً First-Come, First-Served (FCFS).

تعتبر أبسط خوارزميات جدولة CPU التي تم تطويرها وهي خوارزمية غير قابلة للاستياء (غير شفعية) وتُطبق على رتل جهوزية واحد ويتم إسناد الإجرائيات إلى وحدة المعالجة المركزية بحسب ورودها أولاً ولايتم مقاطعتها عند ورود إجرائية أخرى ويتم وضع الإجرائيات الجديدة الواردة في نهاية رتل الجهوزية [21].

- خوارزمية المهمة الأقصر أولاً Shortest Job First(SJF).

وهي خوارزمية غير قابلة للاستيلاء حيث يتم إسناد الإجرائية ذات رشقة المعالج الأقصر إلى المعالج وفي حال كانت إجرائيتين لهما نفس طول رشقة المعالجة يتم استخدام FCFS لاتخاذ القرار.

- خوارزمية الوقت المتبقي الأقصر أولاً Shortest Remaining Time First (SRTF).

وهي خوارزمية قابلة للاستيلاء (شفعية) ويتم إسناد المعالج إلى الإجرائية التي تبقى لها أقل زمن للتنفيذ على المعالج، تتطلب هذه الخوارزمية والخوارزمية التي سبقتها معرفة رشقة المعالج للإجرائية مسبقاً بحيث عندما تصل إجرائية جديدة يتم مقارنة وقتها الإجمالي بالوقت المتبقي لإجرائية الحالية فإذا احتاجت الإجرائية الجديدة لوقت أقل من الإجرائية الحالية، يتم تعليق الإجرائية الحالية وبدء الإجرائية الجديدة.

الجدولة في الأنظمة التفاعلية Interactive systems.

في هذا القسم سوف نلقي نظرة على أبرز خورازميات الجدولة المستخدمة في الأنظمة التفاعلية.

- جدولة Round-Robin (RR)

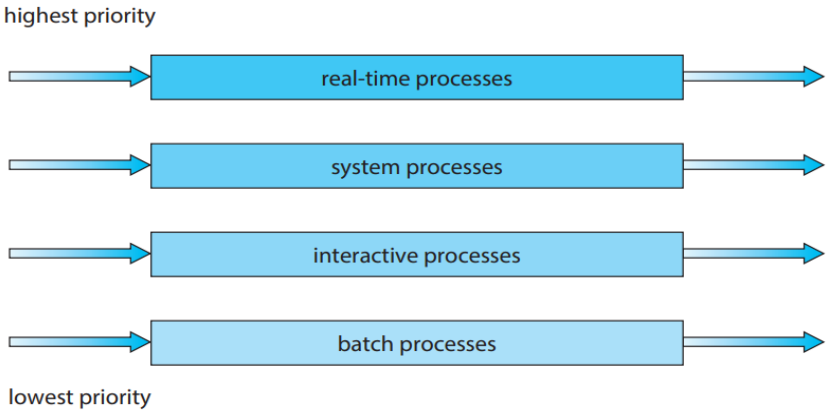
تعتمد على تقسيم زمن المعاج افتراضياً إلى شرائح زمنية تدعى الحصة الزمنية quantum وتكون حركة الإجرائيات التي تحتاج لأكثر من حصة زمنية واحدة لانتهاء تنفيذها ضمن رتل الجهوزية دائرية، في هذه الخوارزمية أطول فترة زمنية يُسمح للإجرائية بالتنفيذ على المعالج في كل مرة تُسلم فيها إليه هو طول الحصة الزمنية وهي خوارزمية غير قابلة للاستيلاء خلال فترة الحصة الزمنية وقابلة للاستيلاء عند انتهاء الحصة.

- الجدولة القائمة على الأولوية Priority Schecduling.

في هذه الخوارزمية يُسند لكل إجرائية أولوية معينة يتم اكتسابها وفق معايير معينة خاصة بنظام التشغيل Operating System (OS) ويتم تخصيص وحدة المعالجة المركزية للإجرائية ذات الأولوية الأعلى أما الإجرائيات المتساوية الأولوية فيتم جدولتها وفق خوارزمية FCFS وعند ورود إجرائية جديدة ذات أولوية أعلى من الإجرائية الحالية فيتم مقاطعة الإجرائية الحالية وإسناد الإجرائية الجديدة إلى وحدة المعالجة المركزية.

- جدولة الرتل متعدد المستويات Multilevel Queue Scheduling.

وتعتمد هذه الجدولة على تقسيم رتل الجهوزية إلى عدة أرتال منفصلة بالاعتماد على نوع الإجرائيات التي تدخل النظام. وقد يتم جدولة الإجرائيات في كل رتل من أرتال الجهوزية (نوع الإجرائيات) باستخدام خوارزمية جدولة مختلفة عن باقي الأرتال مثلاً قد يُخصص رتل الجهوزية الأول لإجرائيات الزمن الحقيقي ويُخصص رتل آخر لإجرائيات النظام ويُخصص رتل آخر للإجرائيات التفاعلية وهكذا كما في الشكل (1-3).



الشكل (1-1): مستويات رتل الجهوزية

كما ويوجد هناك العديد من الخوارزميات التفاعلية الأخرى مثل خورازمية الحصة العادلة Fair-Share Scheduling وخوارزمية جدولة التغذية العكسية متعددة المستويات التي هي موضوع البحث وسنتحدث في فصل لاحق عنها بالتفصيل.

الجدولة في أنظمة الزمن الحقيقي:

نظام الزمن الحقيقي هو النظام الذي يفرض قيوداً زمنية على تدفق المعطيات وعلى التنفيذ وكمثال عن هذه الأنظمة وحدات مراقبة حالات المرضى في غرف العناية المركزة في المشافي، والطيار الآلي و التحكم الآلي في المصانع وغيرها [21].

تصنف أنظمة الزمن الحقيقي إلى أنظمة مرنة Soft Real time systems والتي يكون فيها تجاوز الأزمنة النهائية Deadline أمراً غير مرغوب فيه ولكن مع ذلك مقبول، وأنظمة صارمة Hard Real time systems لا تتسامح أبداً مع تجاوز الأزمنة النهائية.

ويكون هدف المُجدول في أنظمة الزمن الحقيقي التقليدية في تحسين عدد الإجرائيات التي يتنهي تنفيذها قبل أو عند الأزمنة الحرجة الخاصة بها Deadline، وفي جميع أنظمة التشغيل في الزمن الحقيقي RT، يمكن للمجدول استيعاب متطلبات التوقيت عن طريق تعيين أولوية ثابتة أو ديناميكية لكل مهمة من خلال تفويض الاستيلاء Preemption [13].

وقد تكون خوارزميات الجدولة في الزمن الحقيقي إما ساكنة أو ديناميكية حيث يتم اتخاذ قرار الجدولة في الخوارزميات الساكنة قبل بدء تشغيل النظام بينما تُتخذ في الخوارزميات الديناميكية خلال تشغيل النظام.

- خوارزمية Rate-Monotonic Scheduling.

تقوم هذه الخوارزمية بجدولة المهام الدورية باستخدام سياسة أولوية ساكنة Static Priority مع إمكانية استيلاء إجرائية على إجرائية أخرى Preemption، فإذا كانت إجرائية قيد التشغيل وأصبحت إجرائية ذات أولوية أعلى متاحة للتشغيل فهي ستقوم بالاستيلاء على الإجرائية ذات الأولوية الأقل.

يتم تعيين أولوية لكل مهمة دورية بشكل عكسي بناءً على فترتها، كلما كانت الفترة أقصر تزداد الأولوية. وكلما طالت الفترة، انخفضت الأولوية والأساس المنطقي وراء هذه السياسة هو تعيين أولوية أعلى للمهام التي تتطلب وحدة المعالجة المركزية بشكل متكرر.

- جدولة الحصص النسبية Proportional Share Scheduling.

في هذه الجدولة يتم تخصيص عدد من حصص المعالج إلى التطبيقات حيث يمكن للتطبيق الاستحواذ على N حصة من حصص المعالج أي ضمان حصول التطبيق على N/T من إجمالي وقت المعالج.

مثلا: إذا كان إجمالي زمن المعالج هو 100 حصة أي T = 100 وكان سيتم اقتسامها بين ثلاث تطبيقات A,B,C حيث يحصل التطبيق A على 50 حصة والتطبيق B على 15 حصة والتطبيق C على 20 حصة. يجب أن تضمن الخوارزمية حصول كل تطبيق على العدد الذي يحتاجة من الحصص.

يجب أن تعمل هذه الخوارزمية جنباً إلى جنب مع سياسة التحكم في القبول لضمان حصول التطبيق على حصصه من وقت المعالج.

- خوارزمية جدولة الأقل تراخي أولاً Least Laxity First (LLF).

في هذه الخوارزمية يتم تحديد مقدار التراخي (وهو الوقت الذي يمكن أن تتأخر فيه الإجرائية دون تفويت موعدها النهائي Deadline) ويحسب من خلال الفرق بين الموعد النهائي للإجرائية Deadline ووقت التنفيذ المتبقي Remaining execution time. والإجرائية التي يكون لها قيمة التراخي أقل يكون لها أولوية عن التي يكون لها قيمة التراخي أعلى، بحيث يتم إيقاف الإجرائية ذات قيمة التراخي الأعلى واسناد الإجرائية الأقل للمعالجة وتعتبر من الخوارزميات المثالية لمعالجة إجرائيات الزمن الحقيقي RT [5].

وهناك العديد من خوارزميات الجدولة في أنظمة الزمن الحقيقي RTS سنذكر منها خوارزمية جدولة الزمن الحرج أولاً Earliest Deadline First (EDF) والتي تَدخل كمكون رئيسي في بحثنا وسنقوم بشرحها بشكل أوسع في الفصل النظري.

2-1 أهمية البحث

تأتي أهمية هذا البحث كونه يعمل على تحسين خوارزمية من خوارزميات جدولة وحدة المعالجة المركزية وهي خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ بالاعتماد على خوارزمية جدولة في الزمن الحقيقي وهي خوارزمية الزمن النهائي الأقرب أولاً EDF للوصول إلى خوارزمية جديدة مُحسنة بحيث يتم إدخال خوارزمية EDF في مرحلة من مراحل خوارزمية MLFQ بهدف الاستفادة من خصائص وإيجابيات

الخوارزمية الأولى فيما يتعلق بتحسين نسبة نجاح إجرائيات الزمن الحقيقي ومعدل زمن الانتظار وزمن التنفيذ الكلي في الخوارزمية المُقترحة.

1-3 الهدف من البحث

الهدف الذي يسعى هذا البحث للتوصل إليه هو تَحسين خوارزمية جَدولة التَغذية العَكسية متعددة المستويات MLFQ باستخدام خوارزمية الزمن النهائي الأقرب أولاً EDF لأنظمة الزمن الحقيقي بهدف زيادة نسبة نجاح إجرائيات الزمن الحقيقي RT بمقابلتها للمواعيد النهائية Deadlines من خلال زيادة مُعدل النجاح Success Ratio وتقليل كل من معدل زمن الانتظار وزمن التنفيذ الكلي للإجرائيات مقارنتةً بخوارزمية MLFQ.

1-4 المسألة التي يناقشها البحث

يعتبر معيار معدل نجاح إجرائيات الزمن الحقيقي معياراً هاماً وضرورياً لنجاح وظيفة نظام النظام الحقيقي RTS وذلك من خلال زيادة نسبة ملاقاة هذه الإجرائيات للأزمنة النهائية Deadlines علاوةً على تقليل أزمنة انتظار الإجرائيات في أرتال الجهوزية Ready Queues وتقليل أزمنة التنفيذ أيضاً، نناقش في هذا البحث مسألة جدولة مجموعة من العينات المعيارية العشوائية للإجرائيات على مجدول خوارزمية MLFQ ومجدول النسخة المُحسنة سابقاً MLFQ-RT ومجدول الخوارزمية المقترحة MLFQ-EDF بحيث يتم تطبيق عينات الدخل على كل خوارزمية على حدا ومراقبة النتائج على كل من المعايير، نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio ومتوسط زمن الانتظار Average waiting time ومتوسط زمن التنفيذ الكلي Average turnaround time.

نناقش في هذا البحث كيفية تحسين المعايير السابقة من خلال الاستفادة من خاصية خوارزمية EDF بدمجها مع خوارزمية MLFQ وفق الخوارزمية المقترحة.

1-5 الدراسات المرجعية

تم تناول موضوع تحسين خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ في العديد من الأبحاث، اعتمد البعض منها على تحسين معايير زمن التنفيذ الكلي (زمن الإقامة) وزمن الانتظار وزمن الاستجابة

ومعالجة حرمان الإجرائيات في أرتال الجهوزية الأقل أولوية، وأيضاً معايير مدى كفاءة هذه الخوارزمية في معالجة وتخديم إجرائيات الزمن الحقيقي RT وصولاً لأفضل النتائج.

يقترح (Hoganson, K., & Brown, J) **[1]** خوارزمية جديدة تدعى MLFQ-RT لمعالجة إجرائيات الزمن الحقيقي Real Time (RT) ضمن خوارزمية MLFQ من خلال استخدام خوارزمية جدولة الإجرائية الأقصر أولاً Shortest Job First(SJF) في رتل الجهوزية الأعلى أولوية High priority Ready Queue لتخديم إجرائيات الزمن الحقيقي وضمان مقابلتها للمواعيد النهائية Deadlines.

حَسنت الخوارزمية المقترحة من متوسط أزمنة الانتظار ومتوسط أزمنة التنفيذ للإجرائيات ومعالجة حرمان الإجرائيات Starvation من خلال إعادة توجيه جزء من وقت وحدة المعالجة المركزية CPU لتخديم الإجرائيات التي لها أزمنة انتظار كبيرة في أرتال الجهوزية الأقل أولوية.

وَقَدم (Sattar, I., Shahid, M., & Yasir, N) **[5]** خوارزمية جدولة جديدة مُقترحة متعددة المستويات MultiLevel بحيث يتم إدخال الإجرائيات في قائمة List ويتم احتساب الأولوية Priority لكل إجرائية بناءاً على خصائصها وفق معايير زمن الانتظار وزمن التنفيذ والزمن النهائي الأقربDeadline وبعد الاحتساب تَدخل الإجرائيات في الرتل Queue.

يتم تنفيذ كل رتل لفترة زمنية تسمى زمن تنفيذ الرتل Queue Execution Time ليعاد احتساب أولويات الإجرائيات من جديد بعد انقضاء فترة زمن الرتل وتتميز الخوارزمية المقترحة بإمكانية استيعاب إجرائيات الزمن الحقيقي RT بإعطاء إجرائيات الزمن الحقيقي أولويات عالية عن بقية الإجرائيات.

كما يقوم (AbdAlGhani, N., Mohamad, M., Atef, M., & Amgad, K) **[3]** بتعديل عمل خوارزمية جدولة التغذية العكسية متعددة المستوريات MLFQ من خلال بناء مجدول مؤلف من عدة أرتال جهوزية Ready Queues بحيث يتم استخدام خوارزمية Round Robin (RR) في كل رتل من الأرتال السابقة.

يتم احتساب قيمة حصة الشريحة الزمنية quantum في كل رتل وفق: ضرب وسيط رشقات وحدات المعالجة المركزية CPU Bursts المرتبة تصاعدياً في كل رتل جاهزية بعامل رقم مستوى رتل الجهوزية وبالتالي زيادة قيمة الشريحة الزمنية دليل على انخفاض أولوية الإجرائيات في هذا الرتل Queue، أظهرت النتائج تحسناً

ملحوظاً في متوسط زمن التنفيذ الكلي Avg turnaround time ومتوسط زمن الانتظار Avg waiting time.

قام (Thombare, M., Sukhwani, R., Shah, P., Chaudhari, S., & Raundale, P) **[7]** بتطبيق تعديل على خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ باستخدام الشريحة الزمنية الديناميكية Dynamic time quantum في الرتل الأول الأعلى أولوية للمجدول وهو من نوع Round Robin (RR) لتخديم الإجرائيات ذات الرشقات القصيرة في الرتل الأول واستخدام خوارزمية المهمة الأقصر أولاً Shortest job first(SJF) في الرتل الثاني والثالث للمجدول ويتم حساب الشريحة الزمنية وفق 2n/3 حيث n عدد الإجرائيات المتبقية في رتل الجهوزية الأول وقيمة الشريحة الزمنية للرتل الثالث هي قيمة أكبر رشقة CPU لأكبر إجرائية متبقية.

بالمقارنة مع خوارزمية MLFQ ذات الشريحة الزمنية الثابتة Static فقد حَسنت الخوارزمية التي تستخدم الشريحة الديناميكية من متوسط زمن التنفيذ الكلي Avg turnaround time ومتوسط زمن الانتظار Avg waiting time.

كما **قام** (Mulder, D., Ssempala, J., Walton, T., Parker, B., Brough, S., Bush, S., Boroojerdi S., & Tang, J) **[14]**بدراسة تأثير حجم الشريحة الزمنية quantum على خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ وخوارزمية الجدولة الدائرية RR من خلال مجموعة من التجارب باستخدام محاكي نظام تشغيل افتراضي VM، وتبين أن خوارزمية الجدولة تتأثر بشكل كبير بحجم الشريحة الزمنية المستخدم والنسبة بين القيم الكمومية الأصغر والأكبر. في حين أن القيم الصغيرة للشريحة الزمنية تكون أكثر فعالية عند الاخذ بعين الاعتبار زمن الانتظار WT وزمن التنفيذ TT.

ويقدم (Kashif, M., Nawaz, R., Rulaningtyas, R., Iqbal, S., & Zulfiqar, A) **[15]** نسخة مُحسنة وفعالة عن خوارزمية جدوة التغذية العكسية متعددة المستويات MLFQ، من خلال تقسيم المجدول إلى ثلاثة أرتال جهوزية Ready queues واستخدام مفهوم الشريحة الزمنية الديناميكية بدلاً من الشريحة الثابتة. بحيث يتم استخدام خوارزمية RR في الرتل الأول الأعلى أولوية بشريحة زمنية معينة تضمن أن جميع الإجرائيات

ذات الرشقات الزمنية الأقل من قيمة Quantum يتم تنفيذها في الرتل الأول، بينما الإجرائيات ذات الرشقات الأعلى من قيمة الشريحة يتم تحريكها إلى إلى الرتل الثاني وحساب متوسط الرشقات الزمنية للإجرائيات في الرتل الثاني.

يتم إسناد ناتج الاحتساب لمتوسط الرشقات لقيمة الشريحة quantum للرتل الثاني الإجرائيات في الرتل الثاني ذات الرشقات الأقل يتم تنفيذها بينما الإجرائيات ذات الرشقات الأعلى من قيمة الشريحة الزمنية المحسوبة فيتم تحريكها إلى الرتل الثالث وتتحدث قيمة الشريحة الزمنية بشكل تلقائي، بينما الإجرائيات المتبقية في الرتل الثالث فيتم تخديمها وفق خوارزمية FCFS.

يضمن المجدول السابق تخفيض حالة مجاعة أو تأخير الإجرائيات ذات الرشقات الطويلة والتي تُحرم لفترة طويلة من المعالجة وكما يُلاحظ تحسن جيد في معدل انتظار الإجرائيات الوسطي WT ومعدل زمن التنفيذ للإجرائيات TT مقارنة من النسخة ذات قيم الشرائح الثابتة من الخوارزمية.

استعرضنا في هذا الفصل مقدمة عن الجدولة في الأنظمة الحاسوبية وتوضيحها وشرحها ومقدمة عن خوارزميات الجدولة في الأنظمة المتنوعة وأهميتها وأهدافها والمعايير الأساسية لتقييم خوارزميات الجدولة، كما وعرضنا دراسات مرجعية سابقة تطرقت لتحسين خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ بطرق وتعديلات متنوعة وصولاً لأفضل النتائج المرجوة من هذه الخوارزمية.

الفصل الثاني: الدراسة النظرية

ذكرنا سابقاً أن واحدة من خوارزميات الجدولة في الأنظمة التفاعلية هي خوارزميات جدولة الرتل متعدد المستويات، وَأحد أشكال هذه الخوارزميات هي جدولة رتل التغذية العكسية متعدد المستويات.

2-1 خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ

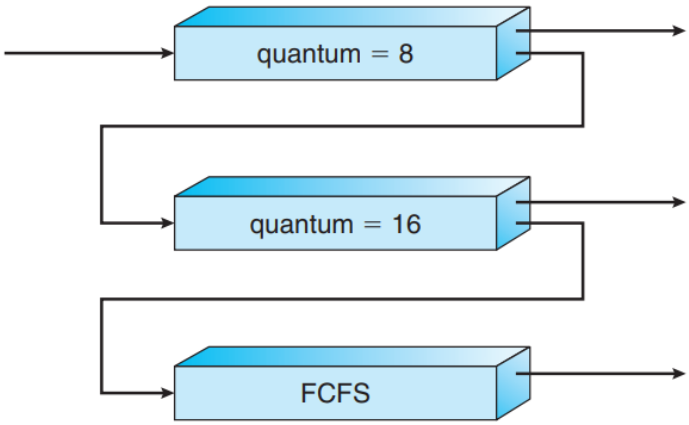
تفترض هذه الخوارزمية أن رتل الجهوزية يُقسم إلى عدة أرتال تُرتب بحيث يكون للرتل الأعلى الأولوية الأعلى والرتل الأدنى الأولوية الأخفض ويتم توزيع الإجرائيات التي يتم الاعتراف بها بين الأرتال وفق خصائصها مثلاً، يتم إسناد إجرائيات الزمن الحقيقي إلى الرتل الأعلى والإجرائيات التفاعلية إلى الرتل التالي بينما الإجرائيات الدفعية إلى الرتل الأدنى.

تضمن خوارزمية جدولة رتل التغذية العكسية متعدد المستويات تحقيق التوازن بين سرعة الاستجابة والعدالة في تخديم الإجرائيات حيث تُخدَّم الإجرائيات ذات الأولوية العالية بطريقة تجعلها أقل عرضة للتأخير من الإجرائيات ذات الأولوية الأخفض ومع ذلك تبقى الإجرائيات ذات الأولوية المنخفضة قادرة على التنفيذ بشكل دوري مما يضمن تلقيها للخدمة في النهاية [8].

تتميز خوارزمية MLFQ بأن الإجرائيات تكون قادرة على الانتقال بين أرتال الجهوزية، حيث تكمن الفكرة في إسناد الإجرائيات وفقاً لخصائها إلى عدة أرتال جهوزية في أنه إذا استهلكت إجرائية معينة الكثير من وقت وحدة المعالجة المركزية، فسيتم نقلها إلى رتل جهوزية له أولوية أقل ولايتم تنفيذ أي إجرائية في رتل جهوزية ذو أولوية أقل طالما توجد إجرائيات في الأرتال الأعلى أولوية [4].

على سبيل المثال الإجرائيات المنتظرة في رتل الجهوزية الثاني لايتم تنفيذها حتى يكون رتل الجهوزية الأول الأعلى أولوية فارغاً وهكذا تباعاً ضمن تسلسل أرتال الجهوزية في المجدول. بالإضافة إلى ذلك فقد يتم نقل الإجرائية التي تنتظر لفترة طويلة في رتل جهوزية ذو أولوية أقل إلى رتل آخر ذو أولوية أعلى. مما يمنع من حدوث حرمان للإجرائيات في أرتال الجهوزية الدنيا الأقل أولوية [19].

يوضح الشكل (2-1) خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ المؤلفة من ثلاثة أرتال جهوزية Ready queues.



الشكل (2-1): خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ

وترتبط خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ بمجموعة من المعاملات:

1. عدد أرتال الجهوزية Ready Queues.
2. خوارزمية الجدولة المستخدمة في كل رتل جهوزية Scheduling Algorithm.
3. الطريقة المُستخدمة لتحديد متى يتم نقل الإجرائية إلى رتل جهوزية ذو أعلى أولوية.
4. الطريقة المُستخدمة لتحديد متى يتم نقل الإجرائية إلى رتل جهوزية ذو أقل أولوية.
5. الطريقة المُستخدمة لتحديد أي رتل جهوزية queue يجب أن تدخل إليها الإجرائية الواجب تَخديمها.

2-1-1 آلية عمل خوارزمية MLFQ

يتم توليد مجموعة الإجرائيات عشوائياً وبحيث يكون لكل إجرائية الخصائص التالية:

1. معرف الإجرائية Process_id
2. زمن الوصول Arrival_time
3. رشقة المعالج Burst_time
4. نوع الإجرائية (إجرائية زمن حقيقي – إجرائية عادية) Process_type

وفي حالة إجرائيات الزمن الحقيقي يُضاف إلى الخصائص السابقة خاصية الزمن النهائي Deadline.

ويتم ترتيب الإجرائيات حسب تسلسل وصولها إلى النظام كما ويتم حساب زمن التنفيذ الكلي (الإجمالي) للإجرائيات Total Execution Time (TET) وهو مجموع رشقات المعالج لكل الإجرائيات.

تهيئة مُجدول خوارزمية MLFQ

يتكون مجدول خوارزمية MLFQ المستخدم من (**5**) أرتال جهوزية Ready Queues (q1...q5) وفق الأولويات من الأعلى إلى الأسفل حيث يكون للرتل الأولى q1 الأولوية الأعلى وللرتل q5 الأولوية الأخفض.

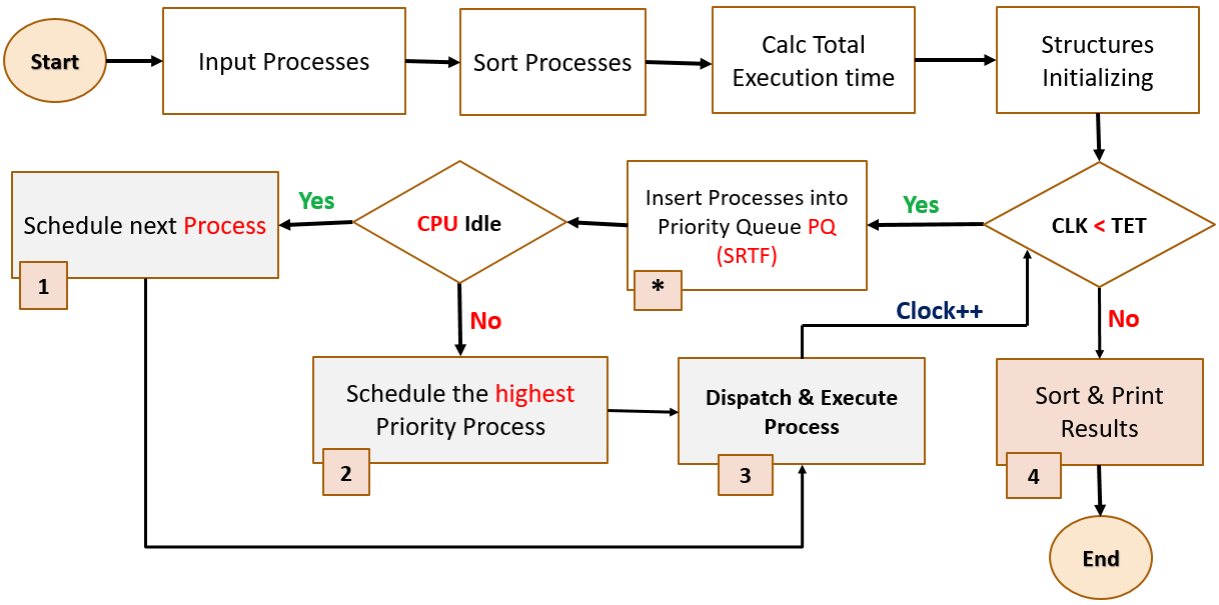
تُجدول إجرائيات الرتل q1 وفق خوارزمية الزمن المتبقي الأقصر أولاً Shortest Remaining Time First (SRTF) وذلك لإعطاء الأولوية Priority للإجرائيات ذات زمن التنفيذ المتبقي الأقصر عن بقية الإجرائيات بحيث اذا وَصلت إجرائية إلى رتل الجهوزية الأول وكان الزمن المتبقي لها أو زمن تنفيذها(رشقة وحدة المعالجة المركزية) أقصر من الإجرائية الحالية قيد التنفيذ Current Process فسوف تُقاطع الإجرائية الحالية وتُسند الإجرائية الجديدة لوحدة المعالجة المركزية.

وتُستخدم خوارزمية الجدولة الدائرية Round Robin (RR) لجدولة الإجرائيات في الأرتال من q2 إلى q5 بشريحة زمنية Time Quantum (tq) ثابتة لكل رتل وهي تُحسب وفق العلاقة التالية:

$$tq = (2)^{number\ of\ queue} \qquad (1)$$

وبهذه الطريقة تُعطى الإجرائيات القصيرة أولوية على الإجرائيات الطويلة مما يحقق مفهوم الأولوية الذي تتميز به خوارزمية MLFQ.

يوضح الشكل (2-2) مخطط عمل خوارزمية MLFQ.



الشكل (2-2): مخطط عمل خوارزمية MLFQ

CLK: دورة الساعة الحالية لـ CPU.

TET: زمن التنفيذ الكلي للإجرائيات Total Execution Time.

مرحلة إدخال الإجرائيات إلى أرتال الجهوزية

عند كل وحدة زمنية من زمن المعاجة الكلي TET يتم إدخال الإجرائيات التي وصلت عند هذه الوحدة الزمنية رتل الجهوزية الأول q1 الذي جدول وفق خوارزمية SRTF ويتم ترتيب الإجرائيات وفق الزمن المتبقي لها وهو يكون مساوياً لطول رشقة المعالج Burst_time (B-time) للإجرائية المدخلة لأول مرّة إلى هذا الرتل.

يتم في المرحلة التالية اختبار حالة وحدة المعالجة المركزية CPU فيما اذا كانت في حالة خمول (Idle) أم لا ليتم الانتقال إلى المرحلة رقم 1 أو المرحلة رقم 2 في الشكل.

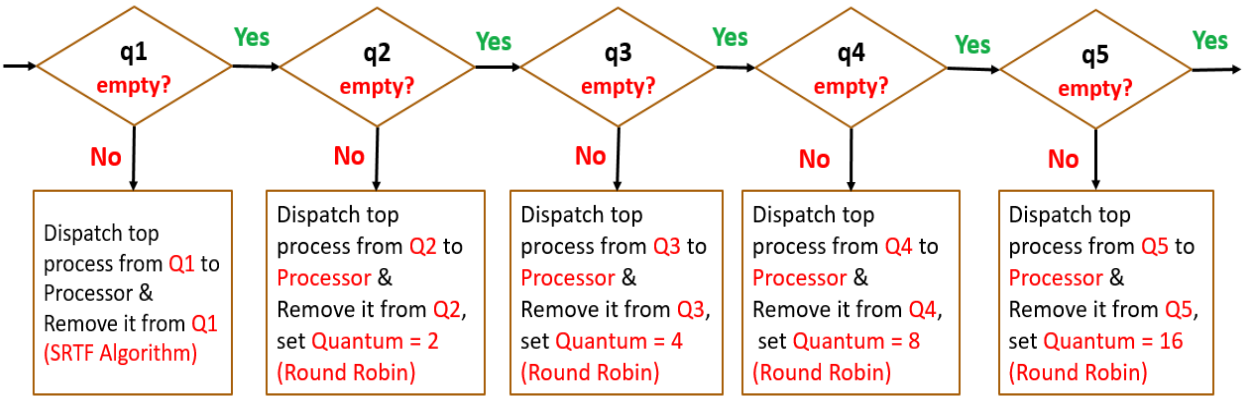
مرحلة اختيار الإجرائية المناسبة واسنادها إلى وحدة المعالجة المركزية

في حال كانت وحدة المعالجة المركزية خاملة (Idle) يتم التحقق من الرتل الأول الأعلى أولوية q1 المُجدول باستخدام خوارزمية SRTF فيما إذا كان فارغاً أم لا، اذا كان ليس فارغاً يتم إخراج الإجرائية الأعلى من الرتل q1 واسنادها للمعالج للتنفيذ.

أما اذا كان فارغاً فيتم التحقق من الرتل الثاني q2 المجدول باستخدام خوارزمية RR بشريحة زمنية tq = 2 في حال كان غير فارغ يتم إخراج الإجرائية الأمامية الأولى من الرتل الثاني q2 واسنادها للمعالج للتنفيذ.

عندما يكون الرتل الثاني فارغاً فيتم التحقق من الرتل الثالث q3 المُجدول وفق خوارزمية RR بشريحة زمنية tq = 4. ويتم إخراج الإجرائية الأمامية الأولى من الرتل الثالث واسنادها للمعالج، وفي حال q3 فارغاً نتحقق من الرتل الرابع q4 المُجدول وفق RR بشريحة زمنية tq = 8 في حال كان غير فارغ فيتم إخراج الإجرائية الأمامية منه واسنادها للمعالج، أما اذا كان فارغاً فيتم التحقق من رتل الجهوزية الأخير q5 وفق خوارزمية RR بشريحة زمنية tq = 16، في حال كان غير فارغ يتم إخراج الإجرائية الأمامية من الرتل واسنادها للمعالج.

في الشكل (2-3).



الشكل (2-3): مرحلة اختيار الإجرائية المناسبة واسنادها إلى وحدة المعالجة المركزية

مرحلة اختيار الإجرائية الأعلى أولوية واسنادها إلى وحدة المعالجة المركزية

عندما تكون وحدة المعالجة المركزية مشغولة Busy يتم تحديد الإجرائية الحالية قيد التنفيذ Current Process ورتل الجهوزية Ready queue الذي خرجت منه الإجرائية الحالية ويتم تمييز الحالات الآتية للإجرائية الحالية Current Process:

- في حال كانت الإجرائية الحالية في الرتل الأول الأعلى أولوية q1.

الإجرائية الحالية من رتل الجهوزية الأول q1 المُجدول وفق خوارزمية SRTF وكان هذا الرتل يحتوي على إجرائيات أخرى بنفس الوقت (ليس فارغاً)، في هذه الحالة يتم مقارنة الزمن المتبقي للإجرائية الحالية Remaining time (R-time) مع الزمن المتبقي للإجراء الأعلى في الرتل q1 وفق الشرط التالي:

$$R - time\ of\ Current\ Process > R - Time\ of\ Top\ Process\ in\ q1 \qquad (2)$$

اذا كان الزمن المتبقي للإجرائية الحالية أصغر من الزمن المتبقي للإجرائية الأعلى في q1، نُخرج الإجرائية الحالية من المعالج وندخلها في رتل الجاهزية الثاني q2 المجدول باستخدام خوارزمية جدولة RR ويتم إسناد الإجرائية الأعلى في الرتل q1 للمعالج.

- في حال كانت الإجرائية الحالية في الرتل الثاني q2.

الإجرائية الحالية من رتل الجهوزية الثاني q2 المُجدول وفق خوارزمية Round Robin (RR) بشريحة زمنية tq = 2 وكان الرتل الأول q1 غير فارغ فيتم إخراج الإجرائية الحالية من المعالج وإدخالها في رتل الجهوزية الثالث q3 وإسناد الإجرائية الأعلى في الرتل q1 للمعالج للتنفيذ.

- في حال كانت الإجرائية الحالية من الرتل الثالث q3.

الإجرائية الحالية من رتل الجهوزية الثالث q3 المُجدو وفق خوارزمية RR بشريحة زمنية tq = 4 وكان الرتل الأول q1 غير فارغ فيتم إخراج الإجرائية الحالية من المعالج وإدخالها في رتل الجهوزية الرابع q4 وإسناد الإجرائية الأعلى في الرتل q1 للمعالج للتنفيذ.

أما في حال كان الرتل الأول q1 فارغاً والرتل الثاني q2 غير فارغ فيتم الاستيلاء على الإجرائية الحالية وإخراجها من المعالج وإدخالها في رتل الجهوزية الرابع q4 وإسناد الإجرائية في مقدمة الرتل q2 للمعالج للتنفيذ إلى المعالج لتنفذ زمنية tq = 2.

- في حال كانت الإجرائية الحالية في الرتل الرابع q4.

الإجرائية الحالية من رتل الجهوزية الرابع المُجدول باستخدام خوارزمية Round Robin (RR) بشريحة زمنية tq = 8 وكان الرتل الأول لأعلى أولوية q1 غير فارغ، يتم إخراج الإجرائية من المعالج وإدخالها في رتل الجهوزية الخامس q5 وإسناد إجرائية الرتل q1 الأعلى أولوية للمعالج للتنفيذ.

واذا كان الرتل q1 الأعلى أولوية فارغاً وكان الرتل الثاني q2 ليس فارغاً فيتم مقاطعة الإجرائية الحالية وإدخالها في رتل الجهوزية الخامس q5 وإسناد الإجرائية من مقدمة الرتل q2 للمعالج ليتم تنفيذها بشريحة زمنية tq = 2.

وفي حال كانت الأرتال q1 والرتل q2 فارغة وكان الرتل الثالث q3 ليس فارغاً فيتم مقاطعة الإجرائية الحالية وإدخالها في رتل الجهوزية الخامس q5 وإسناد الإجرائية من مقدمة الرتل q3 للمعالج ليتم تنفيذها بشريحة زمنية tq = 4.

- في حال كانت الإجرائية الحالية في الرتل الرابع q5.

الإجرائية الحالية من رتل الجهوزية الخامس الأقل أولوية Lowest Priority المُجدول باستخدام خوارزمية Round Robin (RR) بشريحة زمنية tq = 16 وكان الرتل الأول لأعلى أولوية q1 غير فارغ، يتم إخراج الإجرائية من المعالج وإعادتها إلى ذيل رتل الجهوزية الخامس q5 وإسناد إجرائية الرتل q1 الأعلى أولوية للمعالج للتنفيذ.

واذا كان الرتل q1 الأعلى أولوية فارغاً وكان الرتل الثاني q2 ليس فارغاً فيتم مقاطعة الإجرائية الحالية وإعادة إدخالها في ذيل رتل الجهوزية الخامس q5 وإسناد الإجرائية من مقدمة الرتل q2 للمعالج ليتم تنفيذها بشريحة زمنية tq = 2.

وفي حال كانت الأرتال q1 والرتل q2 فارغة وكان الرتل الثالث q3 ليس فارغاً فيتم مقاطعة الإجرائية الحالية وإعادة إدخالها في ذيل رتل الجهوزية الخامس q5 وإسناد الإجرائية من مقدمة الرتل q3 للمعالج ليتم تنفيذها بشريحة زمنية tq = 4.

الحالة الأخيرة عندما تكون الأرتال q1 والرتل q2 و q3 فارغة وكان الرتل الرابع q4 ليس فارغاً فيتم مقاطعة الإجرائية الحالية وإعادة إدخالها في ذيل رتل الجهوزية الخامس q5 وإسناد الإجرائية من مقدمة الرتل q4 للمعالج ليتم تنفيذها بشريحة زمنية tq = 16.

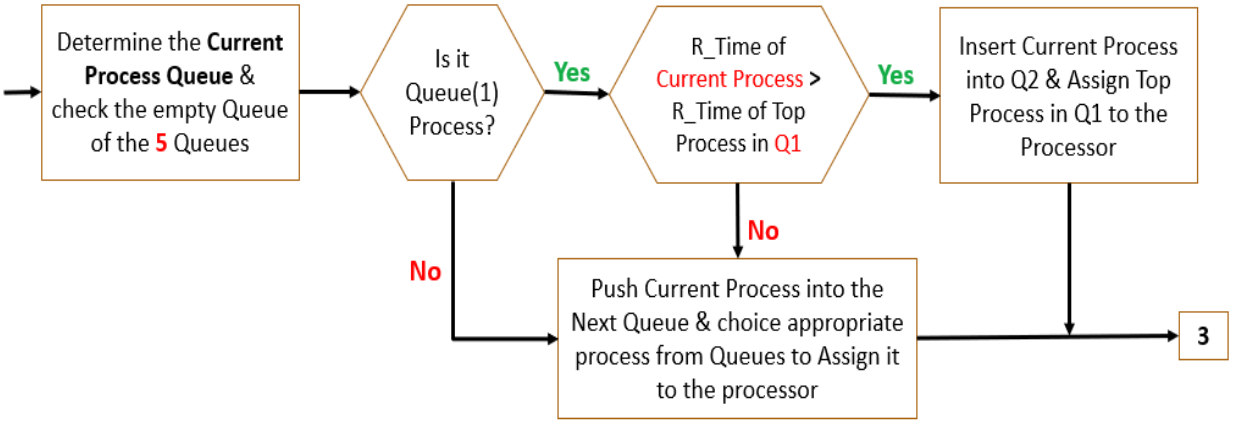
ويوضح الشكل (2-4) هذه المرحلة بحيث:

R_T: الزمن المتبقي للإجرائية Remaining Time.

Cur_pr: الإجرائية الحالية في وحدة المعالجة المركزية Current Process.

q1_pr: إجرائية من الرتل q1 (q1 process).

q1_top_pr: الإجرائية الموجودة في مقدمة رتل الجهوزية q1.



الشكل (2-4): مرحلة اختيار الإجرائية الأعلى أولوية Highest Priority

مرحلة تنفيذ ومعالجة الإجرائيات

خلال دورة الساعة الزمنية الحالية للمعالج clock يتم تنفيذ الإجرائية الحالية Current process في وحدة المعالجة المركزية CPU والتي تم اختيارها وترشيحها من قبل المجدول MLFQ ويتم إنقاص زمن الإجرائية المتبقي Remaining time (R-time) بمقدار 1 للإجرائية الحالية.

وفي حال كانت الإجرائية الحالية في أحدى الأرتال التالية q2 أو q3 أو q4 أو q5 حيث هذه الأرتال مُجدولة وفق خوارزمية Round Robin (RR) وفق الشرائح الزمنية التالية (2-4-8-16) فيتم إنقاص قيمة الشريحة الزمنية المخصصة لإجرائية بمقدار 1 وفق التالي:

- الإجرائية الحالية من رتل الجهوزية الثاني q2 يتم إنقاص الشريحة الزمنية tq2 بمقدار 1.
- الإجرائية الحالية من رتل الجهوزية الثاني q3 يتم إنقاص الشريحة الزمنية tq3 بمقدار 1.
- الإجرائية الحالية من رتل الجهوزية الثاني q4 يتم إنقاص الشريحة الزمنية tq4 بمقدار 1.
- الإجرائية الحالية من رتل الجهوزية الثاني q5 يتم إنقاص الشريحة الزمنية tq5 بمقدار 1.

تقرير انتهاء تنفيذ الإجرائية:

ونميز هنا بين الحالتين التاليتين

1. حالة انتهاء تنفيذ الإجرائية الحالية Current Process Completion

التحقق من انتهاء تنفيذ الإجرائية الحالية في وحدة المعالجة المركزية CPU من خلال إتمام الإجرائية لزمن تنفيذها بحيث تصبح قيمة الزمن المتبقي للإجرائية مساوياً للصفر (R-time = 0)، ويتم تمييز الحالات التالية:

- في حال كانت الإجرائية الحالية من الرتل q1 الأعلى أولوية تصبح حالة وحدة المعاجة المركزية CPU خاملة في نهاية دورة الزمن الحالية.
- في حال كانت الإجرائية الحالية من الرتل q2، تصبح حالة وحدة المعالجة المركزية خاملة ويتم تجديد الشريحة الزمنية للرتل الثاني tq = 2 في نهاية دور الزمن الحالية.

- في حال كانت الإجرائية الحالية من الرتل q3، تصبح حالة وحدة المعالجة المركزية خاملة ويتم تجديد الشريحة الزمنية للرتل الثاني tq = 4 في نهاية دور الزمن الحالية.
- في حال كانت الإجرائية الحالية من الرتل q4، تصبح حالة وحدة المعالجة المركزية خاملة ويتم تجديد الشريحة الزمنية للرتل الثاني tq = 8 في نهاية دور الزمن الحالية.
- في حال كانت الإجرائية الحالية من الرتل q5، تصبح حالة وحدة المعالجة المركزية خاملة ويتم تجديد الشريحة الزمنية للرتل الثاني tq = 16 في نهاية دور الزمن الحالية.

2. حالة عدم انتهاء تنفيذ الإجرائية الحالية Current Process Unfinished

تكون الإجرائية الحالية غير منتهية بعد، حيث يكون الزمن المتبقي للإجرائية لتُنهي تنفيذها غير مساوي للصفر (R-time ≠ 0)، في حال كانت الإجرائية من رتل الجهوزية الأول q1 المُجدول باستخدام خوارزمية SRTF فسيتم تنفيذ الإجرائية في وحدة CPU حتى تأتي إجرائية أخرى لها زمن متبقي أقصر، في هذه الحالة يتم مقاطعة الإجرائية الحالية وإدخالها في ذيل رتل الجهوزية الثاني q2 وإسناد المعالج للإجرائية الأقصر.

بالنسبة للأرتال من q2 إلى q5 المُجدولة باستخدام الجدولة الدائرية RR وانتهت الشريحة الزمنية المخصصة، نميز الحالات التالية:

- انتهاء الشريحة الزمنية للرتل الثاني tq2 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الثالث q3 وجعل حالة وحدة المعالجة المركزية خاملة في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الثاني tq3 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الرابع q4 وجعل حالة وحدة المعالجة المركزية خاملة في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الثاني tq4 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذي الرتل الخامس q5 وجعل حالة وحدة المعالجة المركزية خاملة في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الثاني tq5 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإعادة إدخالها في ذيل الرتل الخامس q5 وجعل حالة وحدة المعالجة المركزية خاملة في نهاية دورة الزمن الحالية.

وتُطبق المراحل السابقة في تنفيذ جميع الإجرائيات الواردة إلى النظام والمُجدولة وفق خوارزمية جدولة MLFQ حتى إتمام تنفيذ كل الإجرائيات.

2-1-2 معايير تقييم الأداء وعرض النتائج

يُقيم أداء خوارزمية MLFQ من خلال المعايير الثلاثة التالية:

1. متوسط زمن الانتظار Average waiting time
2. متوسط زمن التنفيذ الكلي Average turnaround time
3. نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio

يتم حساب زمن انتظار الإجرائية Waiting time من خلال طرح زمن وصول الإجرائية Arrival time من زمن انتهاء الإجرائية Finish time مطروحاً منه طول الإجرائية (رشقة المعالج) Burst time وفق المعادلة التالية:

$$Waiting\ time = (Finish\ time\ - Arrival\ time) - Burst\ time \qquad (3)$$

وزمن التنفيذ الكلي للإجرائية Turnaround time من خلال طرج زمن وصول الإجرائية Arrival time من زمن انتهاء الإجرائية Finish time، كما في المعادلة التالية

$$Turnaround\ time = Finish\ time\ - Arrival\ time \qquad (4)$$

ويُحسب متوسط أزمنة الانتظار لكل الإجرائيات المُجدولة في النظام من خلال جمع قيم أزمنة انتظار كل إجرائية منها وتقسيمها إلى عدد الإجرائيات الكلي في النظام وفق المعادلة التالية:

$$Average\ waiting\ time\ (Avg\ wt) = \frac{\sum_{i=1}^{n} wt_i}{\mathrm{n}} \qquad (5)$$

حيث n هو عدد الإجرائيات في المجدول.

و i عداد متزايد على عدد الإجرائيات.

wt_i زمن انتظار للإجرائية رقم i.

ويُحسب متوسط أزمنة التنفيذ الكلية لكل الإجرائيات المُجدولة في النظام من خلال جمع قيم أزمنة تنفيذ كل إجرائية منها وتقسيمها إلى عدد الإجرائيات الكلي في النظام وفق المعادلة التالية:

$$Average\ turnaround\ time\ (Avg\ tt) = \frac{\sum_{i=1}^{n} tt_i}{\mathrm{n}} \qquad (6)$$

حيث n هو عدد الإجرائيات في المجدول.

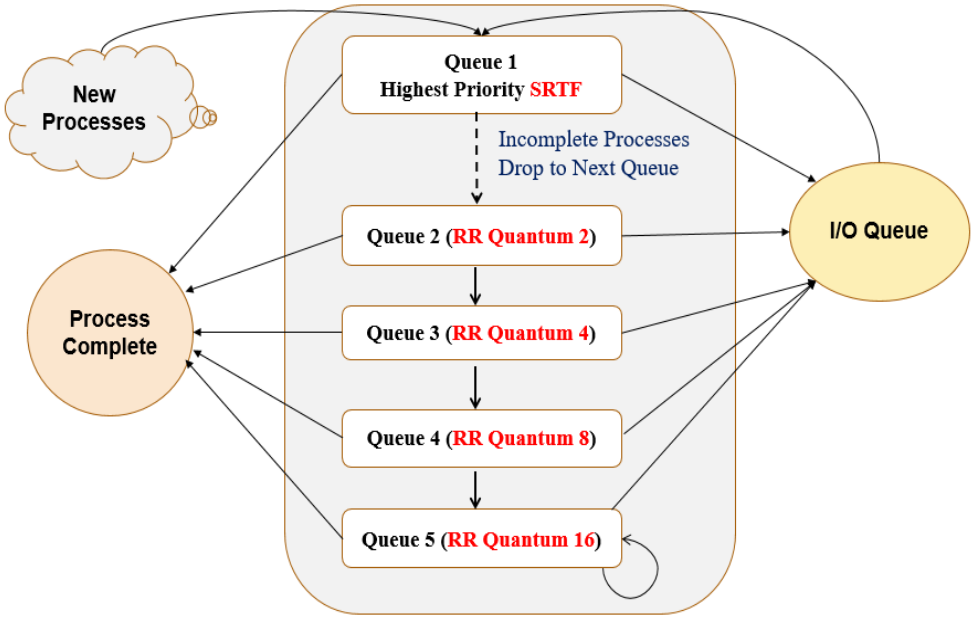
و i عداد متزايد على عدد الإجرائيات.

tt_i زمن التنفيذ الكلي للإجرائية رقم i.

أما إجرائيات الزمن الحقيقي Real Time Processes والتي يكون لها قيمة زمن نهائي Deadline، يتم حساب معدل نجاح هذه الإجرائيات Success Ratio من خلال تقسيم عدد الإجرائيات التي قابلت زمنها النهائي (إجرائيات الزمن الحقيقي التي اكتمل تنفيذها قبل فوات موعدها النهائي) إلى عدد إجرائيات الزمن الحقيقي الكلية في النظام وفق المعادلة التالية:

$$Success\ Ratio = \frac{Count\ of\ Real\ Time\ Processes\ Meet\ Deadline}{Count\ of\ Real\ Time\ Processes} \qquad (7)$$

تشرح الفقرة السابقة عمل مجدول خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ من خلال تقسيم رتل الجهوزية Ready Queue إلى خمس أرتال مرتبة حسب أولوياتها من الأعلى إلى الأدنى وكيفية إدخال الإجرائيات إلى المجدول وتحريك الإجرائيات بين الأرتال وفق المعايير المحددة وانتهاء تنفيذ الإجرائيات، ويوضح الشكل) مخطط الخوارزمية:



الشكل (2-5): مخطط الخوارزمية MLFQ

2-1-3 مثال

يُراد جدولة مجموعة من 10 إجرائيات باستخدام خوارزمية MLFQ بحيث يكون معدل نجاح إجرائيات الزمن الحقيقي Success Ratio أكبر مايمكن و متوسط زمن الانتظار ومتوسط زمن التنفيذ الكلي أصغر مايمكن.

ويتم توليد الإجرائيات عشوائياً بحيث يكون لها الخصائص التالية:

- زمن وصول يتراوح من (1-100).
- رشقة معالج تتراوح من (1-16).
- معرف فريد للإجرائية يتزايد تلقائياً
- نوع الإجرائية Process_type وهي متغير بولياني يأخذ قيمة true في حال كانت الإجرائية إجرائية زمن حقيقي وإلا يأخذ قيمة false.

- الزمن النهائي Deadline لإجرائية الزمن الحقيقي وهو قيمة عشوائية مساوية لزمن وصول الإجرائية مضافاً له طول الإجرائية بالإضافة لقيمة عشوائية تتراوح بين 1 و100 وتُحسب من أجل كل إجرائية وفق المعادلة التالية:

$$Deadline = Arrival\ time + Burst\ time + Rand(1 - 100) \qquad (8)$$

ويبين الجدول مجموعة الإجرائيات المراد جدولتها باستخدام الخوارزمية MLFQ حيث:

Process Id: Pid

Arrival_time: A_time

Burst_time: B_time

Process_type: P_type

قيمة الموعد النهائي للإجرائية Deadline.

يوضح الجدول (2-1) خصائص عينات الدخل لخوارزمية MLFQ

الجدول (0-1): خصائص عينات الدخل لخوارزمية MLFQ

Input Random Standard DataSet (MLFQ)				
Pid	**A_time**	**B_time**	**P_type**	**Deadline**
1	58	11	False	0
2	20	4	False	0
3	9	2	False	0
4	51	6	True	94
5	64	8	True	88
6	22	4	True	111
7	56	11	False	0
8	72	4	True	122
9	50	14	True	109
10	71	9	True	100

يمثل الجدول السابق خصائص 10 إجرائيات مولدة عشوائياً يُراد جدولتها وفق خوارزمية MLFQ بهدف تحسين نسبة النجاح لإجرائيات الزمن الحقيقي Success Ratio وتقليل كل من متوسط زمن الانتظار Average waiting time ومتوسط زمن التنفيذ الكلي Average turnaround time للإجرائيات العادية وبعد تطبيق الخوارزمية على الإجرائيات كانت النتائج كالتالي:

```
  MLFQ Parameters & Criteria :

min_arrival_time        : 9
max_completion_time     : 113
Average Turnaround Time : 20.50
Average Waiting Time    : 13.20
RT_counter              : 6
RT_Meet_Deadline        : 5
Success Ratio           : 0.83
```

تُظهر النتائج أول زمن وصول بين الإجرائيات min arrival time وآخر زمن انتهاء بينها max completion time ومتوسط زمن الانتظار Average Waiting Time ومتوسط زمن التنفيذ الكلي Average Turnaround Time، وعدد إجرائيات الزمن الحقيقي الكلي RT_counter وعدد إجرائيات الزمن الحقيقي التي قابلت الزمن النهائي Deadlines ونسبة نجاح إجرائيات الزمن الحقيقي Success Ratio، يُظهر الجدول (2-2) خرج تنفيذ خوارزمية MLFQ على مجموعة الإجرائيات السابقة وفق المعايير التالية:

زمن انتهاء الإجرائية Finish time ونرمز له F_time.

زمن انتظار الإجرائية Waiting time ونرمز له W_time.

زمن تنفيذ الإجرائية Turnaround time ونرمز له T_time.

الجدول (0-2): أزمنة الإجرائيات بعد تنفيذ خوارزمية MLFQ

Output of MLFQ Algorithm			
Pid	**F_time**	**W_time**	**T_time**
1	111	42	53
2	24	0	4
3	11	0	2
4	57	0	6
5	77	5	13
6	28	2	6
7	68	1	12
8	82	6	10
9	107	43	57
10	113	33	42

بتحليل نتائج تطبيق خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ، تُظهر تحسناً فيما يخص نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio ومتوسط زمن انتظار الإجرائيات Average Waiting Time في CPU ومتوسط زمن التنفيذ الكلي Average Turnaround Time ولكن تسعى الخوارزمية المُقترحة في البحث لتحسين هذه المعايير وهذا ما سنراه في الفقرة الخاصة بالخوارزمية المُحسنة الجديدة.

2-2 خوارزمية جدولة التغذية العكسية متعددة المستويات المُحسنة سابقاً MLFQ-RT

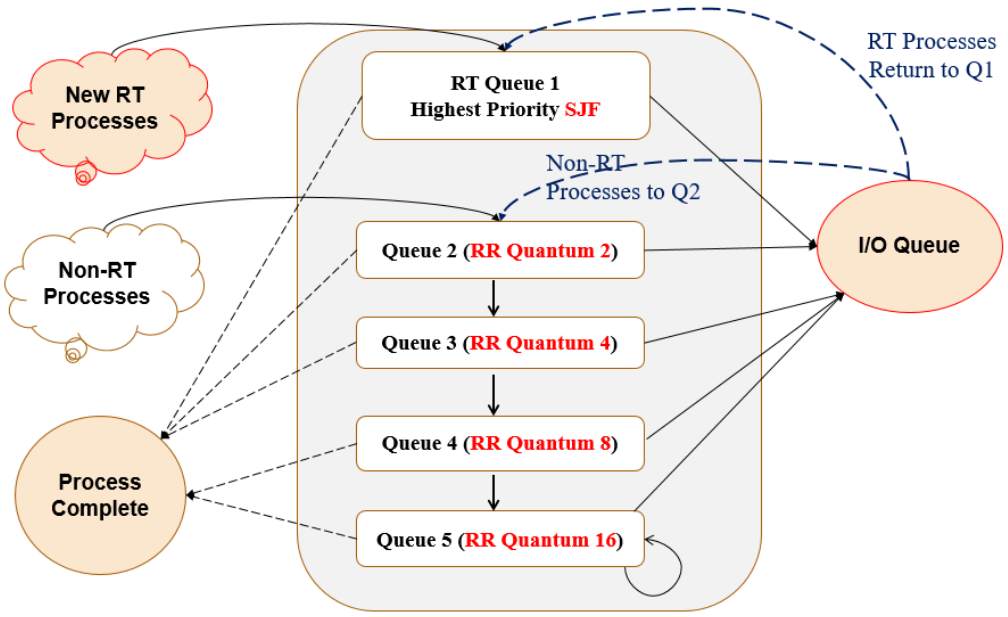
تم اقتراح هذه الخوارزمية في الورقة البحثية [1] كتحسين عن خوارزمية جدولة MLFQ التقليدية لمعالجة وتخديم إجرائيات الزمن الحقيقي Real Time Processes بحيث يتم إعطاء أولوية لهذه الإجرائيات عن بقية الإجرائيات في المجدول وتحسين نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio بمقابلتها للمواعيد النهائية Deadlines، علاوةً عن تحسين متوسط زمن انتظار Average Waiting Time ومتوسط زمن التنفيذ الكلي Average Turnaround Time.

تُصنف خوارزمية MLFQ-RT الإجرائيات الواردة إلى المجدول إلى نوعين من الإجرائيات، النوع الأول هو إجرائيات الزمن الحقيقي Real Time Processes والنوع الثاني هو الإجرائيات العادية Non Real Time Processes.

ويضمن هذا التصنيف إعطاء أولوية أعلى High Priority لإجرائيات الزمن الحقيقي من خلال إدخال فقط إجرائيات الزمن الحقيقي الواردة إلى المُجدول في رتل الجهوزية q1 الأعلى أولوية، بينما يتم إدخال بقية الإجرائيات الأخرى العادية في رتل الجهوزية الثاني q2، بحيث لايمكن تنقيل إجرائيات الزمن الحقيقي من رتل الجهوزية الأعلى q1 إلى الأرتال الأخرى وتبقى هذه الإجرائيات في الرتل q1 حتى تتم معالجتها وتخديمها، بخلاف آلية عمل خوارزمية MLFQ التقليدية والتي تسمح لإجرائيات أي رتل من الأرتال بالتحرك ضمن باقي الأرتال وصولاً لرتل الجهوزية الأقل أولوية q5.

في هذه الخوارزمية MLFQ-RT تبقى إجرائيات الزمن الحقيقي RT ضمن الرتل الأول الأعلى أولوية حتى تُنهي تنفيذها، وتَدخل باقي الإجرائيات العادية Non RT في الرتل q2 ولا يتم تخديم أي إجرائية في الأرتال q2 أو q3 أو q4 أو q5 حتى يُصبح الرتل الأعلى أولوية q1 المُخصص لإجرائيات الزمن الحقيقي فارغاً، في هذه الحالة يمكن تخديم باقي الإجرائيات في الأرتال الأخرى وتنقيل الإجرائيات الغير مكتملة التنفيذ وصولاً لرتل الجهوزية q5 الأدنى أولوية.

عند ورود إجرائية زمن حقيقي RT يتم إدخالها في رتل الجهوزية الأول الأعلى أولوية q1 وجدولتها باستخدام خوارزمية جدولة الزمن الأقصر أولاً Shortest Job First (SJF)، بحيث تُعطى أولوية أعلى High Priority للإجرائيات ذات الطول الأقصر Burst Time، وفي حال كانت إجرائية قيد التنفيذ ضمن وحدة المعالجة المركزية CPU ووردت إجرائية للرتل q1 لها لها طول أقصر فسيتم مقاطعة الإجرائية الحالية Current Process وإسناد الإجرائية الجديدة الأقصر للتنفيذ، مع ملاحظة منع تنقيل إجرائيات الرتل q1 نزولاً لباقي الأرتال الشكل (2-6).



الشكل 6-0): مخطط آلية عمل خوارزمية MLFQ-RT

تتبع خوارزمية MLFQ-RT المُحسنة سابقاً مجموعة من الخطوات في جزء منها مشابهة لخورزمية MLFQ التقليدية ولكن تُوجد اختلافات متعلقة بإعطاء أولوية عالية High Priority لإجرائيات الزمن الحقيقي Real Time (RT) بحيث يتم تصنيف الإجرائيات كما في الشكل إلى نوعين:

1. إجرائيات الزمن الحقيقي Real Time:

يتم إدخال هذه الإجرائيات في الرتل الأول الأعلى أولوية q1 وتُخدم وفق خوارزمية جدولة الزمن الأقصر أولاً SJF مع وضع شرط عدم إمكانية تنقيل إجرائيات الزمن الحقيقي من الرتل q1 إلى بقية الأرتال نزولاً للمحافظة على أولوية عالية لهذه الإجرائيات.

2. إجرائيات عادية Non Real Time:

تُدخل هذه الإجرائيات في رتل الجهوزية الثاني q2 وتُخدم وفق خوارزمية Round Robin بشرائح زمنية (2-4-8-16) ويمكن تحريك هذه الإجرائيات عند انتهاء شرائحها الزمنية في الأرتال q3 و q4 وq5 للمافظة

على آلية عمل خوارزمية MLFQ، وبالتالي تحقيق أولوية أعلى لإجرائيات الزمن الحقيقي في رتل الجهوزية الأول q1.

2-2-1 آلية عمل خوارزمية MLFQ-RT

تتضمن المراحل التالية آلية عمل خوارزمية MLFQ-RT المقترحة سابقاً في الورقة البحثية [1]، وتتبع مجموعة من المراحل الموصحة في البنود التالية:

تهيئة مجدول خوارزمية MLFQ-RT

يتكون مجدول MLFQ-RT من 5 أرتال جهوزية مرقمة من 1 إلى 5 يُجدول كل منها باستخدام خوارزمية جدولة.

1- رتل الجهوزية الأول q1:

يستخدم خوارزمية الزمن الأقصر أولاً Shortest Job First (SJF)، بحيث يتم إدخال إجرائيات الزمن الحقيقي RT فقط إليه ليتم تخديمها حسب الإجرائية ذات الطول الأقصر أولاً Burst Time ولايتم تنفيذ أي إجرائية في الأرتال الأخرى طالما هناك إجرائية موجودة في الرتل q1، ويتم ترتيب الإجرائيات ضمن q1 وفق طول الإجرائية الأقصر Shortest Burst Time.

2- رتل الجهوزية الثاني q2:

يستخدم خوارزمية Round Robin (RR) بشريحة زمنية tq2 = 2 بحيث تأخذ كل إجرائية قيمة الشريحة وعند انتهاء الشريحة الزمنية لهذه الإجرائية يتم تحريكها نزولاً إلى الرتل الثالث q3، لايمكن تنفيذ أي إجرائية ضمن هذا الرتل حتى يكون رتل الجهوزية الأول q1 فارغاً.

3- رتل الجهوزية الثالث q3:

يستخدم خوارزمية Round Robin (RR) بشريحة زمنية tq3 = 4 بحيث تأخذ كل إجرائية قيمة الشريحة وعند انتهاء الشريحة الزمنية لهذه الإجرائية يتم تحريكها نزولاً إلى الرتل الرابع q4، لايمكن تنفيذ أي إجرائية ضمن هذا الرتل حتى يكون رتل الجهوزية الثاني q2 فارغاً.

4- رتل الجهوزية الرابع q4:

يستخدم خوارزمية Round Robin (RR) بشريحة زمنية tq4 = 8 بحيث تأخذ كل إجرائية قيمة الشريحة وعند انتهاء الشريحة الزمنية لهذه الإجرائية يتم تحريكها نزولاً إلى الرتل الخامس q5، لايمكن تنفيذ أي إجرائية ضمن هذا الرتل حتى يكون رتل الجهوزية الثالث q3 فارغاً.

5- رتل الجهوزية الخامس q5:

يستخدم خوارزمية Round Robin (RR) بشريحة زمنية tq5 = 16 بحيث تأخذ كل إجرائية قيمة الشريحة وعند انتهاء الشريحة الزمنية لهذه الإجرائية يتم إعادة إدخالها في الرتل الخامس q5، لايمكن تنفيذ أي إجرائية ضمن هذا الرتل حتى يكون رتل الجهوزية الرابع q4 فارغاً.

مرحلة إدخال الإجرائيات إلى رتل الجهوزية

في كل وحدة زمنية من زمن المعالجة الكلي Total Execution Time (TET) يتم إدخال إجرائيات الزمن الحقيقي RT الواصلة (زمن وصولها Arrival time مساوياً لوحدة الزمن الحالية) إلى الرتل الأول اللأعلى أولوية q1 المُجدول باستخدام خوارزمية جدولة الزمن الأقصر أولاً SJF بينما يتم إدخال بقية الإجرائيات غير الزمن الحقيقي Non RT إلى الرتل الثاني q2 المُجدول باستخدام خوارزمية Round Robin (RR) بشريحة زمنية tq2 = 2.

يتم ترتيب الإجرائيات الواردة إلى رتل اجهوزية الأول q1 بحسب طول الإجرائية الأقصر Burst Time.

مرحلة اختيار الإجرائية واسنادها إلى وحدة المعالجة المركزية في حالة خمول Idle

عندما تكون حالة وحدة المعالجة المركزية خاملة CPU Idle يتم التحقق من أرتال الجهوزية جميعها بدءاً من رتل الجهوزية الأول الأعلى أولوية q1 فيما اذا كان فارغاً Empty أو يحتوي على إجرائيات مُنتظرة.

اذا كان الرتل q1 المخصص لإجرائيات الزمن الحقيقي Real Time Processes غير فارغ يتم إخراج الإجرائية الأعلى منه وإسنادها إلى وحدة المعالجة للتنفيذ وتصبح حالة وحدة المعالجة المركزية مشغولة Busy ويتم إعطاء قيمة لمؤشر دلالة أن الإجرائية تتبع للرتل الأول الأعلى أولوية q1.

وإذا كان الرتل q1 فارغاً فيتم اختبار الرتل الثاني q2 التي تُدخل فيه الإجرائيات الأخرى العادية Non Real Time فيما إذا كان فارغاً أو لا، إذا لم يكن فارغاً يتم إخراج الإجرائية الأعلى منه وإسنادها للمعالج للتنفيذ وإعطاء قيمة لمؤشر دلالة أن الإجرائية تتبع للرتل الاني q2.

وإذا كان الرتل q2 فارغاً، يتم التحقق من q3 فيما اذا كان فارغاً أم لا، وإسناد الإجرائية الأعلى منه لتنفيذ في حالة لم يكن فارغاً. يتم أيضاً التحقق من رتل الجهوزية الآخريين q4 وq5 تباعاً بحيث يكون q4 أكثر أولوية من q5، ويتم اسناد الإجرائية المناسبة للمعالجة ضمن CPU بحسب أولوية هذه الإجرائية.

تمثل المرحلة التالية حالة العمل لوحدة المعالجة لمركزية.

مرحلة اختيار الإجرائية الأعلى أولوية Highest Priority في حالة انشغال Busy

في كل دورة ساعة أو وحدة زمنية من زمن التنفيذ الكلي TET يأتي نهج الخوارزمية MLFQ-RT في اختبار حالة ورود إجرائية على المُجدول حيث يتم التحقق من كون الإجرائية هي إجرائية زمن حقيقي RT أو إجرائية عادية Non RT، ويتم التحقق من جميع الحالات التالية:

- في حال كانت الإجرائية الحالية في رتل الجهوزية q1 المخصص لإجرائيات RT

تُشرح هذه الحالة بأن الإجرائية الحالية في وحدة المعالجة المركزية مصدرها الرتل الأول q1 المُجدول وفق خوارزمية SJF وكانت قد وردت إجرائية أخرى على الرتل q1 في هذه الأثناء فسيتم مقارنة طول الإجرائية الحالية مع طول الإجرائية الواردة

$$Burst\ time\ (Current\ Process) > Burst\ Time\ (\ Top\ Process\ in\ q1) \qquad (9)$$

إذا كان طول الإجرائية الحالية قيد التنفيذ أصغر من طول أعلى إجرائية في q1، تبقى الإجرائية الحالية قيد التنفيذ أما اذا كان طول الإجرائية الحالية أكبر من طول أعلى إجرائية في q1 فسيتم مقاطعة الإجرائية الحالية وإعادة إدخالها إلى ذيل الرتل الأول q1 (نهج MLFQ-RT)، ولايمكن تنقيل إجرائيات الرتل الأول إلى بقية الأرتال. ومن ثم إسناد الإجرائية الأعلى في الرتل q1 للتنفيذ ضمن المعالج.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الثاني q2

عندما يكون رتل الجهوزية الأول q1 غير فارغ، فإنه يتم مقاطعة الإجرائية الحالية ويتم إسناد الإجرائية الأعلى في الرتل q1 للمعالجة بينما تُخدل الإجرائية احالية في الرتل q3 نزولاً.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الثالث q3

يوجد رتلي جهوزية سابقين لهما أولوية أعلى من q3 وهما q1 و q2 ففي حال وجود إجرائية منتظرة في إحدى الرتلين السابقين، يتم مقاطعة الإجرائية الحالية وإدخالها في الرتل الرابع q4 وإسناد الإجرائية الأعلى أولوية للتنفيذ ضمن CPU.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الرابع q4

يوجد ثلاث أرتال جهوزية Ready Queues تسبق في الأولوية الرتل q4 وهم q1 و q2 و q3 ففي حال وجود إجرائية مُنتظرة في إحد الأرتال السابقة، يتم مقاطعة الإجرائية الحالية وإدخالها في الرتل الرابع q5 وإسناد الإجرائية الأعلى أولوية للتنفيذ ضمن CPU.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الرابع q5

يوجد أربع أرتال جهوزية Ready Queues تسبق في الأولوية الرتل q5 وهم q1 وَ q2 وَ q3 وَ q5 ففي حال وجود إجرائية مُنتظرة في إحد الأرتال السابقة، يتم مقاطعة الإجرائية الحالية وإعادة إدخالها في الرتل الخامس q5 وإسناد الإجرائية الأعلى أولوية High Priority من بين الأرتال السابقة للتنفيذ ضمن CPU.

مرحلة تنفيذ ومعالجة الإجرائيات

في هذه المرحلة وبعد تحديد مصدر الإجرائية الحالية من أي رتل جهوزية وردت، يتم إنقاص قيمة الزمن المتبقي للإجرائية الحالية (Reamaining Time (R-time بمقدار القيمة 1 والتي تمثل دورة الساعة المنتهية.

واذا كان مصدر الإجرائية الحالية من إحدى أرتال الجهوزية q2 أو q3 أو q4 أو q5 المجدولة وفق خوارزمية Round Robin (RR) فسيتم أيضاً إنقاص قيمة الشريحة الزمنية المخصصة (Time Quantum (tq بمقدار 1 حسب قيمة الشريحة لكل رتل جهوزية.

تقرير انتهاء تنفيذ الإجرائية:

ونميز هنا الحالتين التاليتين:

1. حالة انتهاء تنفيذ الإجرائية الحالية Current Process Completion

نناقش حالة انتهاء تنفيذ الإجرائية الحالية R-time = 0، تُصبح وحدة المعالجة المركزية في حالة خمول Idle في نهاية دورة الساعة الحالية للإجرائية منتهية التنفيذ. بالإضافة إلى أنه في حال كانت الإجرائية المنتهية تنتمي لأحد أرتال الجهوزية Ready Queues المُجدولة وفق خوارزمية Round Robin (RR) فسيتم تجديد قيمة الشريحة الزمنية المخصصة للإجرائية وفق الآتي:

تجديد قيمة الشريحة الزمنية للرتل الثاني tq2 = 2.

تجديد قيمة الشريحة الزمنية للرتل الثالث tq3 = 4.

تجديد قيمة الشريحة الزمنية للرتل الرابع tq4 = 8.

تجديد قيمة الشريحة الزمنية للرتل الخامس tq5 = 16.

أما في حال لم ينتهي تنفيذ الإجرائية الحالية Current Process فيتم مناقشة عدة حالات بناءاً على مصدر الإجرائية الحالية قيد التنفيذ في المعالج.

2. حالة عدم انتهاء تنفيذ الإجرائية الحالية Current Process Unfinished

تختبر الخوارزمية حالة انتهاء الشريحة الزمنية Time Quantum للإجرائية الحالية قيد التنفيذ في وحدة المعالجة المركزية CPU، ويتم مناقشة مصدر الإجرائية وقيمة الشريحة المخصصة لها والتي تُجدول وفق خوارزمية Round Robin (RR) بشرائح مختلفة وفق مايلي:

- انتهاء الشريحة الزمنية للرتل الثاني tq2 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الثالث q3 وتكون وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.

- انتهاء الشريحة الزمنية للرتل الثاني tq3 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الرابع q4 وتكون وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الثاني tq4 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الخامس q5 وتكون وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الثاني tq5 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإعادة إدخالها في ذيل الرتل الخامس q5 وتكون وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.

وتُطبق المراحل السابقة في تنفيذ جميع الإجرائيات الواردة إلى النظام والمُجدولة وفق خوارزمية MLFQ-RT حتى إتمام تنفيذ كل الإجرائيات.

2-2-2 معايير تقييم الأداء وعرض النتائج

يُقيم أداء خوارزمية MLFQ-RT من خلال المعايير الثلاثة التالية:

1. متوسط زمن الانتظار Average waiting time
2. متوسط زمن التنفيذ الكلي Average turnaround time
3. نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio

تشرح الفقرة السابقة عمل خوارزمية الجدولة في الورقة البحثية [1] MLFQ-RT من خلال تقسيم رتل الجهوزية Ready Queue إلى خمس أرتال مرتبة حسب أولوياتها من الأعلى إلى الأدنى وكيفية تخصيص الرتل الأول الأعلى أولوية لإجرائيات الزمن الحقيقي RT وباقي الأرتال الأخرى للإجرائيات العادية Non RT مع منع إجرائيات رتل الجهوزية الأول المُجدول وفق خوارزمية SJF من النزول وفق الأرتال الأخرى والسماح بتنقيل الإجرائيات ضمن الأرتال الأخرى والتي تُجدول وفق خوارزمية Round Robin.

3-2-2 مثال

يُراد جدولة مجموعة من 10 إجرائيات باستخدام خوارزمية MLFQ-RT بحيث يكون معدل نجاح إجرائيات الزمن الحقيقي Success Ratio أكبر مايمكن و متوسط زمن الانتظار ومتوسط زمن التنفيذ الكلي أصغر مايمكن.

نستخدم مجموعة العينات السابقة التي تم استخدامها في خوارزمية MLFQ التقليدية، لنقارن معاير تقييم الأداء بين الخوارزميتين MLFQ و خوارزمية MLFQ-RT بالنسبة لنفس مجموعة العينات ويبين الجدول (2-3) مجموعة الإجرائيات المراد جدولتها باستخدام الخوارزمية MLFQ-RT حيث:

.

الجدول (0–3): خصائص عينات الدخل لخوارزمية MLFQ-RT

Input Random Standard DataSet (MLFQ-RT)				
Pid	**A_time**	**B_time**	**Is_Real_Time**	**Deadline**
1	58	11	False	0
2	20	4	False	0
3	9	2	False	0
4	51	6	True	94
5	64	8	True	88
6	22	4	True	111
7	56	11	False	0
8	72	4	True	122
9	50	14	True	109
10	71	9	True	100

عند تطبيق عينات الدخل السابقة من الإجرائيات المولدة عشوائياً على خوارزمية MLFQ-RT المُحسنة سابقاً في الورقة البحثية [1]، يُظهر الجدول (2–4) خرج تنفيذ الخوارزمية MLFQ-RT على العينات السابقة لكل إجرائية وفق المعايير التالية:

زمن انتهاء الإجرائية Finish time ونرمز له F_time.

زمن انتظار الإجرائية Waiting time ونرمز له W_time.

زمن تنفيذ الإجرائية Turnaround time ونرمز له T_time.

في الجدول (0–4 أزمنة الإجرائيات بعد التنفيذ.

الجدول (0-4): أزمنة الإجرائيات بعد تنفيذ خوارزمية MLFQ-RT

Output of MLFQ-RT Algorithm			
Pid	**F_time**	**W_time**	**T_time**
1	113	44	55
2	28	4	8
3	11	0	2
4	87	30	36
5	72	0	8
6	26	0	4
7	108	41	52
8	91	15	19
9	64	0	14
10	81	1	10

يمثل الجدول السابق خصائص 10 إجرائيات مولدة عشوائياً يُراد جدولتها وفق خوارزمية MLFQ-RT المُحسنة سابقاً بهدف تحسين نسبة النجاح لإجرائيات الزمن الحقيقي Success Ratio وتقليل كل من متوسط زمن الانتظار Average waiting time ومتوسط زمن التنفيذ الكلي Average turnaround time للإجرائيات العادية وبعد تطبيق الخوارزمية على الإجرائيات كانت النتائج كالتالي:

```
    MLFQ_RT Parameters & Criteria :

min_arrival_time         : 9
max_completion_time      : 113
Average Turnaround Time  : 20.80
Average Waiting Time     : 13.50
RT_counter               : 6
RTMeetDeadline           : 6
SussessRatio             : 1.00
```

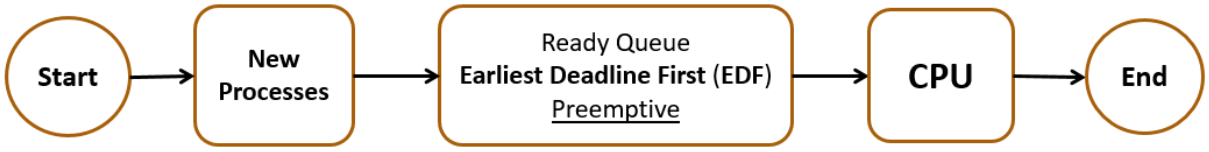
بتحليل نتائج تطبيق خوارزمية جدولة MLFQ-RT، تُظهر تحسناً فيما يخص نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio ومتوسط زمن الانتظار Average Waiting Time في CPU ومتوسط زمن التنفيذ الكلي Average Turnaround Time ولكن تسعى الخوارزمية المُقترحة في البحث لتحسين هذه المعايير وهذا ما سنراه في الفقرة الخاصة بالخوارزمية المُقترحة الجديدة.

2-3 خوارزمية الزمن النهائي الأقرب أولاً Earliest Deadline First (EDF)

تُعد واحدة من أهم خوارزميات الجدولة في أنظمة الزمن الحقيقي RTS، تُستخدم لجدولة إجرائيات الزمن الحقيقي Real Time Proesses وضمان تنفيذ تلك الإجرائيات قبل فوات مواعيدها النهائية (مقابلتها للحدود الزمنية الحرجة Deadlines). [19]

تعتمد على تعيين أولويات الإجرائيات Priorities في النظام بشكل ديناميكي [16] وفي كل لحظة وفقاً لأزمنتها الحرجة أي أن الإجرائية ذات الزمن الحرج الأبكر أو الأقرب Deadlines هي ذات الأولوية الأعلى والتي سوف يتم تخديمها في الوقت الحالي. كما تعتبر خوارزمية EDF خوارزمية قابلة للاستيلاء Preemptive. تتطلب هذه الخوارزمية من كل إجرائية أن تُفصح عن موعدها النهائي (الزمن الحرج Deadline) عندما تصبح قابلة للتشغيل Runnable.

عند ورود إجرائية إلى المجدول لها قيمة الزمن النهائي (الحرج) أبكر أو أقرب، فسيتم إيقاف الإجرائية الحالية وإسناد الإجرائية ذات الزمن الحرج الأقرب للمعالجة.



الشكل (2-7): خوارزمية الزمن الحرج الأقرب أولاً EDF

تعتبر خوارزمية EDF مثالية من الناحية النظرية theoretically optimal، فمن الناحية النظرية يمكنها جدولة الإجرائيات بحيث يمكن لكل إجرائية تلبية متطلبات الموعد النهائي الخاصة بها وسيكون استخدام وحدة المعالجة المركزية بالطاقة القصوى، ومع ذلك من الناحية الإجرائية من المستحيل تحقيق هذا المستوى من استخدام وحدة المعالجة المركزية بسبب تكلفة التبديل بين السياق بين الإجرائيات ومعالجة المقاطعات.

يمكننا استخدام خوارزمية الزمن الحرج الأقرب EDF لتحسين خوارزمية MLFQ فيما يتعلق بنسبة نجاح إجرائيات الزمن الحقيقي ومتوسط زمن انتظار الإجرائيات ومتوسط زمن التنفيذ الكلي وهذا ما سنراه في الفقرة القادمة شرح الخوارزمية المُقترحة في هذا البحث.

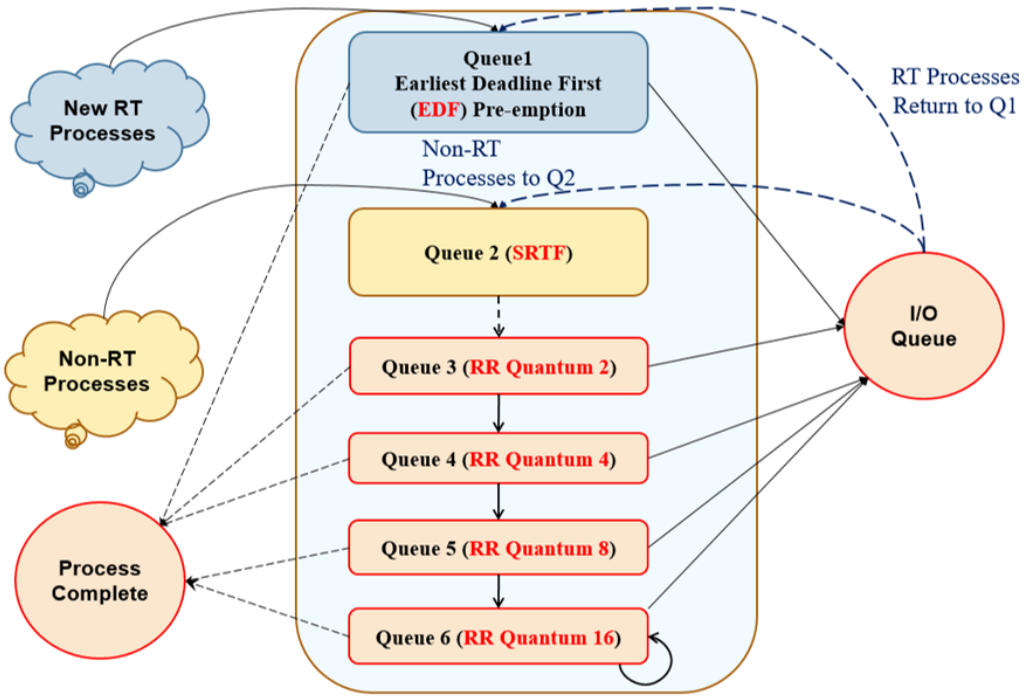
2-4 الخوارزمية المُقترحة MLFQ-EDF

بعد دراسة خوارزمية MLFQ وخوارزمية MLFQ-RT المُقترحة في الورقة البحثية الأولى [1]، وجدنا أن الخوارزمية المقترحة في الورقة البحثية الأولى [1] حَسنت من نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio ومتوسط زمن الانتظار ومتوسط زمن التنفيذ الكلي، ولكن لضمان تنيفذ إجرائيات الزمن الحقيقي ضمن الحدود الزمنية Deadlines يجب استخدام خوارزمية جدولة في الزمن الحقيقي لمعالجة وتخديم إجرائيات الزمن الحقيقي بالشكل الأمثل.

حيث أنه لايتم استخدام خوارزميات جدولة في الزمن الحقيقي لمعالجة إجرائيات الزمن الحقيقي في خوارزمية MLFQ وخوارزمية الورقة البحثية الأولى MLFQ-RT وهذا ينعكس سلبياً على نسبة نجاح إجرائيات الزمن الحقيقي RT بمقابلتها للأزمنة النهائية.

وبعد دراسة خوارزمية جدولة الزمن النهائي الأقرب أولاً Earliest Deadline First (EDF) في القسم النظري، لُوحظ أن هذه الخوارزمية تعتبر مثالية لتخديم إجرائيات الزمن الحقيقي وضمان مقابلتها للمواعيد النهائية وذات كفاءة عالية في تحقيق أعلى استخدام لوحدة المعالجة المركزية.

مما سبق يمكننا استنتاج وتوقع أن إدراج خوارزمية جدولة الزمن النهائي الأقرب أولاً EDF ضمن رتل الجهوزية الأول الأعلى أولوية في خوارزمية التغذية العكسية متعددة المستويات MLFQ وتهيئة الخوارزمية المُقترحة الجديدة من شأنه أن يُحسن نسبة نجاح إجرائيات الزمن الحقيقي ويقلل متوسط زمن الانتظار ومتوسط زمن التنفيذ الكلي الإجرائيات الشكل (2-8).



الشكل (2-8): مخطط آلية عمل الخوارزمية المُقترحة MLFQ-EDF

2-4-1 آلية عمل الخوارزمية المقترحة MLFQ-EDF

وتتبع الخوارزمية المُحسنة الجديدة مجموعة من الخطوات التالية:

تهيئة مجدول الخوارزمية المقترحة

يتم توليد مجموعة الإجرائيات عشوائياً وبحيث يكون لكل إجرائية الخصائص التالية:

1. معرف الإجرائية Process_id
2. زمن الوصول Arrival_time
3. رشقة المعالج Burst_time
4. نوع الإجرائية (إجرائية زمن حقيقي – إجرائية عادية) Process_type

وفي حالة إجرائيات الزمن الحقيقي يُضاف إلى الخصائص السابقة خاصية الزمن النهائي Deadline.

ويتم ترتيب الإجرائيات حسب تسلسل وصولها إلى النظام وفي حال تساوي أزمنة الوصول لإجرائيتين أو أكثر إلى المجدول يتم أخذ معيار الموعد النهائي الأصغر Deadline لترتيب الإجرائيات في المجدول ويتم حساب زمن التنفيذ الكلي (الإجمالي) للإجرائيات Total Execution Time (TET).

يتألف مُجدول MLFQ-EDF من 6 أرتال جهوزية Ready Queues مرتبة حسب الأولوية من الرتل q1 وصولاً للرتل q6، تُقسم الإجرائيات إلى قسمين هما إجرائيات الزمن الحقيقي RT وإجرائيات عادية Non RT بحيث يتم إدخال فقط إجرائيات الزمن الحقيقي في الرتل الأول الأعلى أولوية q1 وتُدرج بقية الإجرائيات العادية في الرتل q2.

تُصنف خوارزميات الجدولة المُستخدمة في كل رتل جاهزية وفق التالي:

- خوارزمية الجدولة المستخدمة في الرتل q1 هي خوارزمية الزمن النهائي الأقرب أولاً EDF وذلك لمعالجة إجرائيات الزمن الحقيقي RT ضمن الرتل الأول.
- خوارزمية الجدولة المُستخدمة في الرتل الثاني q2 هي خوارزمية الزمن المتبقي الأقصر أولاً SRTF وذلك لمعالجة الإجرائيات العادية Non RT.
- خوارزمية الجدولة المستخدمة في الرتل الثالث q3 هي خوارزمية Round Robin (RR) بشريح زمنية مساوية tq3 = 2.
- خوارزمية الجدولة المستخدمة في الرتل الرابع q4 هي خوارزمية Round Robin (RR) بشريح زمنية مساوية tq4 = 4.
- خوارزمية الجدولة المستخدمة في الرتل الخامس q5 هي خوارزمية Round Robin (RR) بشريح زمنية مساوية tq5 = 8.
- خوارزمية الجدولة المستخدمة في الرتل السادس q6 هي خوارزمية Round Robin (RR) بشريح زمنية مساوية tq6 = 16.

يتم إدخال إجرائيات الزمن الحقيقي RT في الرتل الأول q1 ولايمكن تحريكها من هذا الرتل نزولاً إلى باقي الأرتال، حيث تبقى في الرتل الأول وتُخدم وفق خوارزمية EDF ويكون لإجرائيات هذا الرتل الأولوية عن باقي الأرتال الأخرى في الحصول على وحدة المعالجة المركزية CPU.

وفق خوارزمية الزمن النهائي الأقرب أولاً EDF يتم تنفيذ الإجرائية ذات الموعد النهائي الأقرب وفي حال كانت إجرائية قيد التنفيذ ضمن المعالج ووَردت إجرائية أخرى لها موعد نهائي أقرب Deadline فسيتم مقاطعة الإجرائية الحالية قيد التنفيذ وإسناد الإجرائية الأخرى للتنفيذ ضمن المعالج.

كما يتم إدراج الإجرائيات العادية Non RT في رتل الجهوزية الثاني q2 المُجدول وفق خوارزمية SRTF، ليتم تنفيذها ضمن هذا الرتل والإجرائيات التي لاتُكمل تنفيذها يتم تنقيلها إلى الأرتال الأدنى من الرتل q3 وصولاً للرتل q6 باستخدام خوارزمية RR وفق شرائح زمنية مختلفة لكل رتل جاهزية Ready queue.

يتم التعبير عن المخطط التدفقي للخوارزمية المُقترحة وفق الشكل (2-9) بحيث:

CLK: دورة الساعة الحالية لـ CPU.

TET: زمن التنفيذ الكلي للإجرائيات Total Execution Time.

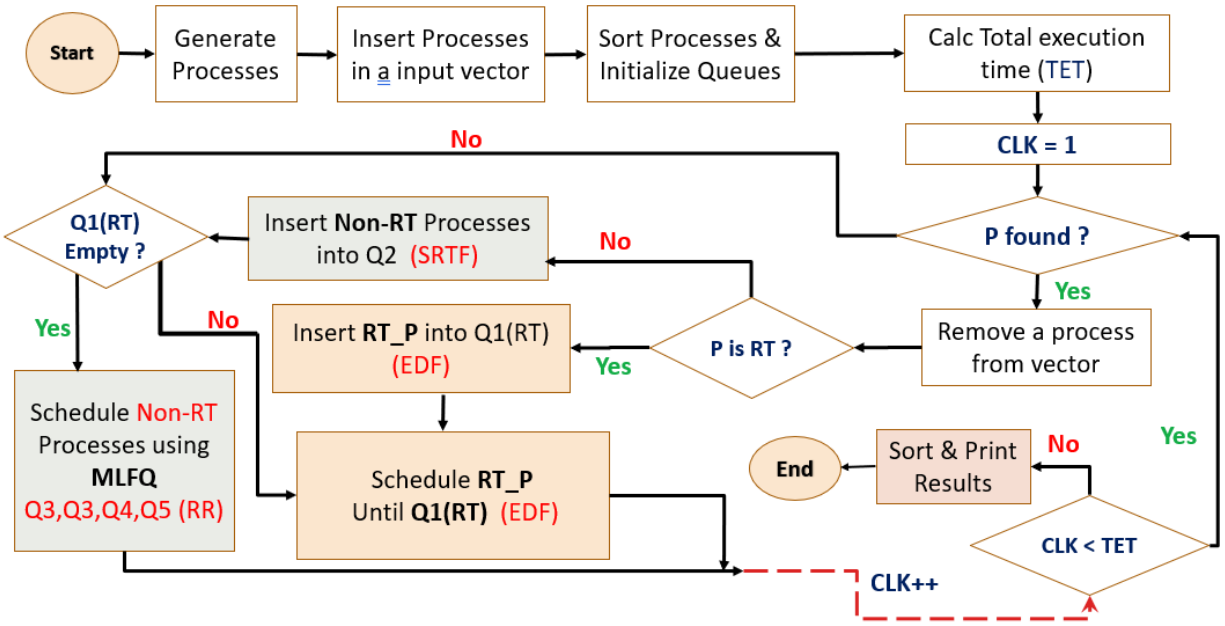
New_pr: وجود إجرائية واصلة تنتظر الجدولة.

RT_pr: إجرائية زمن حقيقي Real time.

Non-RT: إجرائية عادية.

Q1 (RT): رتل الجهوزية الأعلى أولوية المخصص لإجرائيات الزمن الحقيقي.

Q1 (SRTF): رتل الجهوزية الثاني المخصص للإجرائيات العادية وفق خوارزمية SRTF.



الشكل (2-9): المخطط التدفقي للخوارزمية المُقترحة MLFQ-EDF

مرحلة إدخال الإجرائيات إلى أرتال الجهوزية Ready queues

بعد تهيئة مجدول خوارزمية MLFQ-EDF، يتم حساب زمن التنفيذ الكلي Total Execution Time(TET).

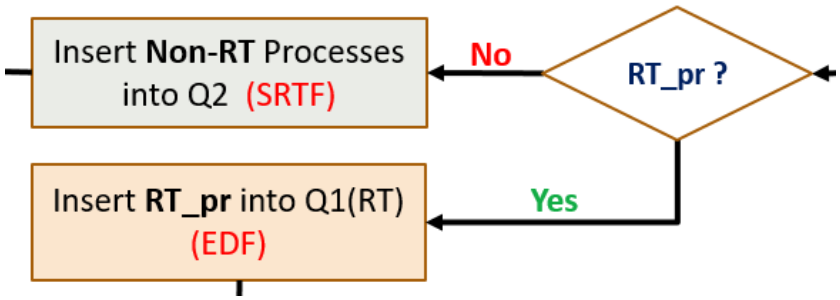
فخلال كل دورة ساعة من زمن المعاجة الكلي يتم اختبار حالة وجود إجرائية واصلة حالية إلى المُجدول (زمن الوصول للإجرائية Arrival Time مساوي لدورة الساعة الحالية)، يتم التحقق من خصائص الإجرائية الواردة حالياً وفق مايلي:

- الإجرائية الواردة حالياً هي إجرائية زمن حقيقي RT من خلال التحقق من وجود قيمة الموعد النهائي للإجرائية Deadline والمؤشر Process_type (Is_Real_Time) والذي يدل أيضاً على أن الإجرائية هي إجرائية زمن حقيقي، فيتم إدخال الإجرائية إلى الرتل الأول q1 الأعلى أولوية المخصص

لإجرائيات الزمن الحقيقي والمُجدول وق خوارزمية الزمن النهائي الأقرب أولاً EDF ووفق ترتيب معين بحسب أقرب موعد نهائي ضمن الرتل q1.

- الإجرائية الواردة حالياً هي إجرائية عادية Non RT فيتم إدخالها مباشرةً في الرتل الثاني q2 المخصص لهذا النوع من الإجرائيات والمُجدول وفق خوارزمية الزمن الأقصر المتبقي أولاً SRTF وترتيب الإجرائيات في الرتل حسب الزمن المتبقي الأقصر Shortest Remaining Time (R-time).

ويوضح الشكل (2-10) مرحلة اختبار خصائص الإجرائية قبل إدخالها في أرتال الجهوزية Ready queues المناسبة:



الشكل (2-10): مرحلة اختبار خصائص الإجرائية قبل إدخالها إلى أرتال الجهوزية

مرحلة اختيار الإجرائية المناسبة واسنادها إلى وحدة المعالجة المركزية CPU

في هذه المرحلة يتم بدايةً التحقق من حالة وحدة المعالجة المركزية فيما اذا كانت بحالة خمول (Idle) أو بحالة انشغال (Busy) ليتم تنفيذ مجموعة من الخطوات التالية وفق:

1. وحدة المعالجة المركزية في حالة خمول (CPU Idle):

يتم التحقق من جميع أرتال الجهوزية Ready queues في المُجدول واختبار فيما اذا كان كل رتل من الأرتال فارغاً أم لا وفق تراتبية معينة تبدأ من الرتل q1 الأعلى أولوية، في حال كان الرتل q1 المخصص لإجرائيات الزمن الحقيقي RT والمُجدول وفق خوارزمية EDF ليس فارغاً فيتم إسناد الإجرائية الأعلى في الرتل q1

للتنفيذ في المعالج والتي يكون لها أقرب موعد نهائي Earliest Deadline بحيث تأخذ أولوية عن باقي الإجرائيات الأخرى الموجودة في الرتل الأول ويتم إخراج الإجرائية من الرتل q1.

عندما يكون الرتل q1 فارغاً يتم اختبار الرتل الثاني q2 فيما اذا كان فارغاً أم لا، فإذا لم يكن فارغاً يتم إخراج الإجرائية الأعلى منه وإسنادها للتنفيذ في وحدة المعالجة المركزية بحيث تكون إجرائيات الرتل الثاني q2 مجدولة وفق خوارزمية SRTF.

بينما الأرتال من (q3…q6) فتكون مُجدول وفق خوارزمية RR بشرائح زمنية مختلفة وفق مايلي:

- عندما يكون الرتل q2 فارغاً يتم إخراج الإجرائية الأمامية الأعلى من الرتل الثالث q3 واسنادها للمعالج للتنفيذ وفق شريحة زمنية tq3 = 2.
- عندما يكون الرتل q3 فارغاً يتم إخراج الإجرائية الأمامية الأعلى من الرتل الرابع q4 واسنادها للمعالج للتنفيذ وفق شريحة زمنية tq4 = 4.
- عندما يكون الرتل q4 فارغاً يتم إخراج الإجرائية الأمامية الأعلى من الرتل الخامس q5 واسنادها للمعالج للتنفيذ وفق شريحة زمنية tq5 = 8.
- عندما يكون الرتل q5 فارغاً يتم إخراج الإجرائية الأمامية الأعلى من الرتل السادس q6 واسنادها للمعالج للتنفيذ وفق شريحة زمنية tq6 = 16.

وفي جميع الحالات السابقة لجميع الأرتال في المجدول عند إسناد الإجرائية للتنفيذ تُصبح حالة وحدة المعالجة المركزية بحالة انشغال CPU Busy.

2. وحدة المعالجة المركزية في حال انشغال (CPU Busy):

عندما تكون وحدة المعالجة المركزية عاملة Busy يتم تحديد الإجرائية الحالية قيد التنفيذ Process Current ورتل الجهوزية queue Ready الذي خرجت منه الإجرائية الحالية فيما اذا كانت الإجرائية هي إجرائية زمن حقيقي RT أو ليست إجرائية عاديةNon RT ويتم تمييز الحالات التالية للإجرائية الحالية Current Process قيد التنفيذ:

- في حال كانت الإجرائية الحالية في رتل الجهوزية q1 الأعلى أولوية المخصص لإجرائيات RT

عندما يكون مصدر الإجرائية الحالية من رتل الجهوزية الأول q1 وبالتالي هي إجرائية زمن حقيقي وكان الرتل الأول ليس فارغاً، فيتم مقارنة الموعد النهائي Deadline للإجرائية الحالية قيد التنفيذ مع الموعد النهائي للإجرائية الأعلى في الرتل q1. وعندما يكون الموعد النهائي للإجرائية الأعلى في q1 أصغر من الموعد النهائي للإجرائية الحالية فيتم مقاطعة الإجرائية الحالية Current Process وإعادة إدخالها في الرتل q1 ومن ولايمكن تحريك إجرائيات الرتل الأول إلى بقية أرتال الجهوزية وفق آلية عمل الخوارزمية المُقترحة في البحث MLFQ-EDF ثم إسناد الإجرائية أأعلى في q1 للتنفيذ في وحدة المعالجة المركزية CPU وتكون المقارنة وفق الشرط التالي:

$$Deadline\ (Current\ Process) > Deadline\ (\ Top\ Process\ in\ q1) \qquad (10)$$

- في حال كانت الإجرائية الحالية من رتل الجهوزية الثاني q2 المخصص للإجرائيات Non RT.

الرتل الثاني مُجدول وفق خوارزمية SRTF، عندما تكون الإجرائية الحالية مصدرها رتل الجهوزية الثاني q2 وكان رتل الجهوزية الأول q1 الأعلى أولوية ليس فارغاً فسيتم مقاطعة الإجرائية الحالية Current وإسناد إجرائية الزمن الحقيقي الأعلى من الرتل q1 للتنفيذ ضمن وحدة الماجة المركزية.

الحالة الثانية: عندما تكون الإجرائية مصدرها الرتل q2 ووردت إجرائية إلى الرتل q2 نفسه ولكن للإجرائية الجديدة لها زمن تنفيذ متبقي أصغر Remaining time فسيتم أيضاً مقاطعة تنفيذ الإجرائية الحالية وإخراجها من المعالج وإدخالها في ذيل رتل الجهوزية الثالث q3، وإسناد الإجرائية الجديدة الواردة إلى q2 للتنفيذ ضمن وحدة المعالجة المركزية.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الثالث q3

الرتل الثالث q3 مُجدول وفق خوارزمية RR بشريحة زمنية مساوية لـ tq3 = 2، فاذا كانت الإجرائية الحالية مصدرها الرتل الثالث ووردت إجرائية للرتل الأول q1 الأعلى أولوية. فسيتم مقاطعة تنفيذ الإجرائية الحالية وإدخالها في الرتل الرابع q4 وإسناد الإجرائية الموجودة في مقدمة الرتل q1 للتنفيذ.

وإذا كان الرتل الثاني q2 ليس فارغاً والرتل الأول q1 فارغاً، فسيتم إيضاً مقاطعة تنفيذ إجرائية الرتل الثالث q3 وإسناد الإجرائية الموجودة في مقدمة الرتل q2 للتنفيذ.

مما يضمن تنقيل الإجرائيات ضمن أرتال الجهوزية في المُجدول.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الرابع q4

الرتل الرابع q4 مُجدول وفق خوارزمية RR بشريحة زمنية مساوية لـ tq4 = 4، وعندما تكون الإجرائية الحالية مصدرها من الرتل الرابع ووردت إجرائية إلى الرتل q1 فسيتم مقاطعة الإجرائية الحالية قيد التنفيذ وإدخالها إلى الرتل الخامس q5 وإسناد الإجرائية الموجودة في مقدمة الرتل q1 للتنفيذ. أو وكان الرتل الثاني q2 ليس فارغاً فسيتم أيضاً مقاطعة الإجرائية الحالية وإدخالها في q5 وإسناد الإجرائية الموجودة في مقدمة الرتل q2 للتنفيذ.

وعندما يكون الرتل q3 ليس فارغاُ، فسيتم مقاطعة الإجرائية الحالية التي مصدرها q4 وإدخالها أيضاً ضمن q5 وإسناد الإجرائية الأمامية الأعلى في الرتل q3 للتنفيذ في CPU.

- في حال كانت الإجرائية الحالية في رتل الجهوزية الخامس q5

الرتل الرابع q5 مُجدول وفق خوارزمية RR بشريحة زمنية مساوية لـ tq5 = 8، وعندما تكون الإجرائية الحالية مصدرها من الرتل الخامس ووردت إجرائية إلى الرتل q1 فسيتم مقاطعة الإجرائية الحالية قيد التنفيذ وإدخالها إلى الرتل السادس q6 وإسناد الإجرائية الأعلى في الرتل q1 للتنفيذ. أو وكان الرتل الثاني q2 ليس فارغاً فسيتم أيضاً إيقاف الإجرائية الحالية وإدخالها في ذيل الرتل q6 وإسناد الإجرائة الأعلى في الرتل q2 للتنفيذ في المعالج.

وعندما يكون الرتل q3 ليس فارغاُ، فسيتم مقاطعة الإجرائية الحالية التي مصدرها q4 وإدخالها أيضاً في ذيل الرتل q5 وإسناد الإجرائية الموجودة في مقدمة الرتل q3 للتنفيذ في CPU.

وأيضاً في حال كان الرتل q4 ليس فارغاُ، فسيتم مقاطعة الإجرائية الحالية التي مصدرها q5 وإدخالها أيضاً في ذيل الرتل q6 وإسناد الإجرائية الموجودة في مقدمة الرتل q4 للتنفيذ في CPU.

- مصدر الإجرائية الحالية من رتل الجهوزية السادس q6

هذا الرتل مُجدول وفق خوارزمية RR بشريحة زمنية مساوية لـ tq6 = 16، وكانت الإجرائية الحالية مصدرها من هذا الرتل فيتم التحقق من الأرتال من q1 إلى q5 فيما اذا كان أحد هذه الأرتال ليس فارغاً، فيتم مقاطعة الإجرائية الحالية وإعادة إدخالها في ذيل الرتل q6 لكونه اخر رتل وأدنى رتل في الأولوية Low Priority في المُجدول المقترح. وإسناد الإجرائية الأعلى أولوية من أكثر الأرتال أولوية بحسب الترتيب للتنفيذ ضمن وحدة المعالجة المركزية.

مرحلة تنفيذ ومعالجة الإجرائيات ضمن CPU

خلال دورة الساعة الزمنية الحالية للمعالج clock يتم تنفيذ الإجرائية الحالية Current Process في وحدة المعالجة المركزية CPU والتي تم اختيارها وترشيحها من قبل مُجدول الخوارزمية المقترحة MLFQ-EDF ويتم إنقاص الزمن المتبقي للإجرائية الحالية Remaining Time (R-time) بمقدار (1) التي تمثل دورة ساعة واحدة. كما يتم إنقاص الشريحة الزمنية Time Quantum المخصصة للإجرائية الحالية بمقدار (1) في حال كان مصدر الإجرائية الحالية من أحد الأرتال التالية q3 أو q4 أو q5 أو q6 المُجدولة وفق خوارزمية Round Robin (RR).

تقرير انتهاء تنفيذ الإجرائية

ونميز هنا بين الحالتين التاليتين:

1. حالة انتهاء تنفيذ الإجرائية الحالية Current Process Completed

يتم التحقق من انتهاء زمن تنفيذ الإجرائية الحالية في وحدة المعالجة المركزية بحيث يُصبح الزمن المتبقي للإجرائية الحالية مساوي للصفر R-time = 0 في نهاية دورة الساعة Clock.

نناقش مجموعة الحالات التي تمثل معرفة رتل الجهوزية Ready queue التي وردت منه الإجرائية الحالية في المعالج وآليه المعالجة المخصصة في كل حالة من الحالات وفق الجدول (0-5):

الجدول (0-5): حالة الإجرائية الحالية وأرتال الجهوزية Ready Queues

الحالة	النتيجة
الإجرائية الحالية من رتل الجهوزية الأول q1 المُجدول وفق خوارزمية EDF ومخصص لإجرائيات الزمن الحقيقي RT	في نهاية دورة الساعة تصبح وحدة المعالجة المركزية بحالة خمول CPU Idle.
الإجرائية الحالية من رتل الجهوزية الثاني q2 المُجدول وفق خوارزمية SRTF وتُدخل إليه الإجرائيات Non RT	في نهاية دورة الساعة تصبح وحدة المعالجة المركزية بحالة خمول CPU Idle.
الإجرائية الحالية من رتل الجهوزية الثالث q3 المُجدول وفق خوارزمية RR بشريحة زمنية tq3 = 2	في نهاية دورة الساعة تصبح وحدة المعالجة المركزية بحاة خمول ويتم تجديد الشريحة الزمنية tq3.
الإجرائية الحالية من رتل الجهوزية الرابع q4 المُجدول وفق خوارزمية RR بشريحة زمنية tq4 = 4	في نهاية دورة الساعة تصبح وحدة المعالجة المركزية بحالة خمول ويتم تجديد الشريحة الزمنية tq4.
الإجرائية الحالية من رتل الجهوزية الخامس q5 المُجدول وفق خوارزمية RR بشريحة زمنية tq5 = 8	في نهاية دورة الساعة تصبح وحدة المعالجة المركزية بحالة خمول ويتم تجديد الشريحة الزمنية tq5.
الإجرائية الحالية من رتل الجهوزية السادس q6 المُجدول وفق خوارزمية RR بشريحة زمنية tq6 = 16	في نهاية دورة الساعة تصبح وحدة المعالجة المركزية بحالة خمول CPU Idle ويتم تجديد الشريحة الزمنية tq6.

أما في حالة عدم انتهاء الإجرائية الحالية Current Process فيتم مناقشة عدة حالات بناءاً عى مصدر الإجرائية الحالية في وحدة المعالجة المركزية CPU.

2. حالة عدم انتهاء تنفيذ الإجرائية الحالية Current Process Incomplete

تقوم الخوارزمية MLFQ-EDF باختبار حالة انتهاء الشريحة الزمنية Time Quantum للإجرائية الحالية قيد التنفيذ في وحدة المعالجة المركزية بعد تحديد مصدر الإجرائية من أي رتل جهوزية ومعرفة قيمة الشريحة الزمنية الكاملة المخصصة لها وفق خوارزمية RR بشرائح معينة وفق مايلي:

- انتهاء الشريحة الزمنية للرتل الثالث tq3 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الرابع q4 وتصبح وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الرابع tq4 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإدخالها في ذيل الرتل الخامس q5 وتصبح وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل الخامس tq5 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإعادة إدخالها في ذيل الرتل الخامس q5 وتصبح حالة وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.
- انتهاء الشريحة الزمنية للرتل السادس tq6 = 0، يتم إخراج الإجرائية الحالية من التنفيذ وإعادة إدخالها في ذيل الرتل الخامس q6 وتصبح حالة وحدة المعالجة المركزية بحالة خمول في نهاية دورة الزمن الحالية.

وتُطبق المراحل السابقة في تنفيذ جميع الإجرائيات الواردة إلى النظام والمُجدولة وفق خوارزمية MLFQ-EDF حتى إتمام تنفيذ كل الإجرائيات.

2-4-2 معايير تقييم الأداء وعرض النتائج

تم اقتراح خوارزمية جديدة تدعى MLFQ-EDF من خلال استخدام خوارزمية من خوارزميات الجدولة في الزمن الحقيقي وهي خوارزمية الزمن النهائي الأقرب أولاً Earliest Deadline First (EDF) ضمن رتل الجهوزية الأول الأعلى أولوية Ready Queue من خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ وتعديل عمل خوارزمية MLFQ من خلال منع إمكانية تنقيل إجرائيات الزمن الحقيقي RT ضمن أرتال المُجدول وتخصيص الرتل الأول لإجرائيات الزمن الحقيقي RT والرتل الثاني وبقية أرتال المُجدول للإجرائيات العادية Non RT، الغاية المرجوة من الخوارزمية المقترحة هي تحسين نسبة نجاح إجرائيات الزمن

الحقيقي Succes Ratio وتقليل كل من متوسط زمن انتظار الإجرائيات Average Waiting Time ومتوسط زمن التنفيذ الكلي للإجرائيات Average Turnaround Time في وحدة المعالجة المركزية CPU.

استعرضنا في هذا الفصل شرح وتوضيح كل من خوارزمية جدولة التغذية العكسية متعدة المستويات MLFQ وتطبيقها في الجدولة والخوارزمية المقترحة في الورقة البحثية الأولى [1] MLFQ-RT وتطبيقها أيضاً في الجدولة.

كما تم شرح وتوضيح الخوارزمية الجديدة المُقترحة في هذا البحث MLFQ-EDF وشرح مراحل وخطوات هذه الخوارزمية وآلية عملها، سيتم شرح طريقة تطبيقها ومقارنة نتائجها ومعايير تقييم الأدء مقارنةً مع الخوارزميات السابقة في فصل التطبيق العملي التالي.

الفصل الثالث: التطبيق العملي

تم في الفصل النظري السابق استعراض ودراسة خوارزمية جدولة التغذية اتعكسية متعددة المستويات MLFQ وتطبيقها في جدولة الإجرائيات. وأيضاً دراسة الخوارزمية MLFQ-RT المُحسنة سابقاً في الورقة البحثية [1] وتطبيقها في الجدولة ثم التعرف على خواررزمية الزمن الحرج الأقرب أولاً EDF، واقتراح خوارزمية جديدة في البحث هي MLFQ-EDF من خلال استخدام خوارزمية EDF ضمن خوارزمية MLFQ وشرح مراحل الخوارزمية المُقترحة.

نستعرض في هذا الفصل التطبيق العملي للخورزمية المقترحة المُحسنة MLFQ-EDF والنتائج العملية لها ونناقش مميزات كل خوارزمية من الخوارزميات السابقة بالإضافة لمقارنة وتحليل نتائج تطبيق كل منها في الجدولة.

3-1 بيئة التطوير وتنجيز الخوارزميات

تم استخدام لغة C++ لكتابة الكود البرمجي واستخدام النمط Windows Platform لتنفيذ المشروع من نوع Console App على بيئة التطوير Visual Studio.

تم تنظيم المشروع في بيئة التطوير ضمن عدة صفوف Classes وبنى معطيات Data Structure وتوابع Functions لإنجاز خوارزمية MLFQ وخوارزمية MLFQ-RT وبناء الخوارزمية المُقترحة MLFQ-EDF.

الصفوف المُستخدمة في بيئة التطوير موضحة في الجدول (3-1):

الجدول (0-1): الصفوف المستخدمة في بيئة التطوير

الترتيب	الصف
1	Process_Data
2	comPare_MLFQ_Time
3	comPare_MLFQ_EDF_Time
4	comPare_MLFQ_RT_Time

1. الصف Process_Data:

يمثل هذا الصف الإجرائية وخصائصها بحيث يكون لكل إجرائية في النظام عدة خصائص تمثل طبيعة هذه الإجرائية وسلوكها وماهيتها وتشمل المتحولات التالية كما في الجدول (3-2):

الجدول (0-2): خصائص الإجرائية ضمن الصف Process_Data

Num	رقم تسلسلي للإجرائية
Pid	معرف خاص بالإجرائية
A_time	زمن وصول الإجرائية
B_time	طول الإجرائية (رشقة CPU)
Priority	أولوية الإجرائية
F_time	زمن انتهاء (إتمام) الإجرائية
R_time	الزمن المتبقي للإجرائية
W_time	زمن انتظار الإجرائية
T_time	زمن تنفيذ الإجرائية
S_time	زمن البدء بتنفيذ الإجرائية
Res_time	زمن الاستجابة
Deadline	الموعد النهائي للإجرائية (الزمن الحرج)
Is_RealTimeProcess	مؤشر للدلالة على إجرائية الزمن الحقيقي RT
Process_Type (P_type)	نوع الإجرائية (زمن حقيقي أو عادية)

2. الصف comPare_MLFQ_SRTF_Time:

يحتوي هذا الصف على تابع يعمل على مقارنة الزمن المتبقي للإجرائيتين R_time اللتان يتم المقارنة بينهما بحيث يعيد قيمة بوليانية Boolean قيمتها True عندما يكون الزمن المتبقي للإجرائية الأولى (البارامتر الأول للتابع) أصغر من الزمن المتبقي للإجرائية الثانية (البارامتر الثاني للتابع).

3. الصف comPare_MLFQ_EDF_Time:

يحتوي هذا الصف على تابع يعمل على مقارنة قيمة الزمن النهائي Deadline للإجرائية الأولى وتمثل (البارامتر الأول) مع قيمة الزمن الحرج للإجرائية الثانية وتمثل (البارامتر الثاني) ويعيد قيمة بوليانية True في حال كان الزمن الحرج للإجرائية الأولى أصغر من الزمن الحرج للإجرائية الثانية

4. comPare_MLFQ_RT_Time:

يحتوي هذا الصف على تابع يعمل على مقارنة قيمة رشقة المعالج للإجرائية الأولى (B_time) وتمثل (البارامتر الأول) مع قيمة رشقة المعالج للإجرائية الثانية وتمثل (البارامتر الثاني) ويعيد قيمة بوليانية True في حال كانت رشقة الإجرائية الأولى أصغر من رشقة الإجرائية الثانية.

بنى المعطيات Data Structure لتقييم معايير الأداء:

يتم تهيئة مجموعة بنى المعطيات والمتغيرات Variables المخصصة لتقييم أداء الخوارزمية المقترحة ومقارنتها مع الخوارزمية الأساسية والخوارزمية المُحسنة سابقاً في الورقة البحثية [1] وتكون وفق الجدول (3-3):

الجدول (0-3): بنى المعطيات والمتغيرات المستخدمة لتقييم أداء الخوارزمية المُقترحة

المتحول	الدلالة
min_arrival_time	زمن الوصول الأصغري للإجرائيات
max_completion_time	زمن الانتهاء الأعظمي للإجرائيات
avg_turnaround_time	متوسط أزمنة التنفيذ للإجرائيات
avg_waiting_time	متوسط أزمنة الانتظار للإجرائيات
Sussess_Ratio	نسبة نجاح إجرائيات الزمن الحقيقي
RT_Meet_Deadline	عدد إجرائيات الزمن الحقيقي التي قابلت الحدود الزمنية الحرجة

ويعرض الجدول (3-4) بنى المعطيات والمتغيرات المُستخدمة في تنفيذ الخوارزميات وبناء المجدولات

الجدول (0-4): بنى المعطيات والمتغيرات المستخدمة في تنفيذ الإجرائيات وبناء المُجدولات

priority_queue	رتل الأولوية وهو بنية خاصة بإمكانها ترتيب الإجرائيات بداخلها فق ترتيب معين يحدد مسبقاً حيث يتم ترتيب الإجرائيات بداخلها وفق الزمن المتبقي الأصغر للإجرائية Remaining time في خوارزمية MLFQ، ووفق رشقة الإجرائية الأقصر Burst time في خوارزمية MLFQ-RT وتُرتب وفق الموعد النهائي أو الزمن الحرج Deadline في الخوارزمية المُقترحة في البحث MLFQ-EDF.
queue	رتل جهوزية يقوم بترتيب الإجرائيات بداخله وفق طريقة FIFO وذلك للتعبير عن خوارزمية Round Robin بحيث تكون يتم تخديم الإجرائية الداخلة إلى الرتل أولاً ونحتاج إلى 4 أرتال ضمن خوارزمية MLFQ
pq_process rq2_process rq3_process rq4_process rq5_process	متغيرات تأخذ قيمة 1 أو 0 للدلالة على مصدر الإجرائية الحالية حيث المتغير pq_process يستخدم للدلالة على أن مصدر الإجرائية هو رتل الأولوية الأول q1 المُجدول وفق خوارزمية SRTF في خوارزمية MLFQ والمُجدول وفق SJF في خوارزمية MLFQ-RT المُحسنة سابقاً، بينما باقي الأرتال الأربعة المتبقية المُرمزة بـ rq تدل على أن مصدر الإجرائية هو من أحد الأرتال الأربعة المُجدولة وفق خوارزمية RR
rt_process	متغير يعد كمؤشر للدلالة على أن الإجرائية الحالية مصدرها رتل الجهوزية الأول q1 المخصص لإجرائيات الزمن الحقيقي Real Time وذلك في خوارزمية MLFQ-EDF المقترحة في البحث
cpu_state	متغير يُعد كمؤشر للدلاة على وضع CPU الحالي بحالة خمول Idle أو انشغال Busy ويأخذ القيمتين 0 أو 1

quantum_rq2 quantum_rq3 quantum_rq4 quantum_rq5	متغيرات لتخزين قيمة الشرائح الزمنية لأرتال الجهوزية المُجدولة وفق خوارزمية Round Robin وتأخذ القيم التالية بالترتيب (2-4-8-16)
current	وهي غرض من الصف Process_Data وتمثل الإجرائية الحالية قيد المعالجة في وحدة المعالجة المركزية ويكون لها جميع خصائص الصف الذي أُشتقت منه.
total_exection_time(tet)	متغير يمثل زمن التنفيذ الكلي للإجرائيات وَيُحسب من خلال جمع قيم رشقات الإجرائيات الداخلة إلى المُجدول حيث أن دورة الساعة clock للمعالجة تتم وفق كل وحدة زمنية من قيمة هذا المتغير tet

ويعرض الجدول (3-5) التوابع المُستخدة لترتيب الإجرائيات ضمن أرتال الجهوزية

الجدول (0-5): التوابع المُستخدمة لترتيب الإجرائيات ضمن أرتال الجهوزية Ready Queues

الترتيب	التابع
1	idsort(const P_d& x, const P_d& y)
2	Numsort(const P_d& x, const P_d& y)
3	arrivalsort_MLFQ_Time(const P_d& x, const P_d& y)
4	arrivalsort_MLFQ_RT_Time(const P_d& x, const P_d& y)
5	arrivalsort_MLFQ_EDF_Time(const P_d& x, const P_d& y)

1. التابع idsort(const P_d& x, const P_d& y)

يُستخدم هذا التابع لترتيب الإجرائيات حسب قيمة معرف الإجرائية ويأخذ بارمتران هما الإجرائيتين اللتان يتم المقارنة بينهما ويعيد القيمة البوليانية True عندما يكون معرف الإجرائية الأولى أصغر من معرف الإجرائية الثانية لكل إجرائيتين يتم المقارنة بينهما.

2. التابع Numsort(const P_d& x, const P_d& y)

يُستخدم هذا التابع لترتيب الإجرائيات حسب رقم الإجرائية ويأخذ بارمتران هما الإجرائيتين اللتان يتم المقارنة بينهما ويعيد القيمة البوليانية True عندما يكون رقم الإجرائية الأولى أصغر من رقم الإجرائية الثانية لكل إجرائيتين يتم المقارنة بينهما.

3. التابع arrivalsort_MLFQ_Time(const P_d& x, const P_d& y)

يُستخدم هذا التابع ضمن مراحل نمذجة خوارزمية جدولة التغذية العكسية متعددة المستويات الأساسية MLFQ وذلك لترتيب الإجرائيات بحسب زمن وصولها للنظام Arrival Time، حيث يأخذ التابع إجرائيتين للمقارنة بينهما من خلال مقارنة زمن وصول كل منهما بالآخر ويعيد القيمة المنطقية True في حال كان زمن وصول الإجرائية الأولى أصغر من ومن وصول الإجرائية الثانية إلى النظام. وعند تساوي أزمنة الوصول بين الإجرائيتين يتم مقارنة الزمن المتبقي Remaining Time لكل إجرائية مع الأخرى وعند التساوي أيضاً يتم مقارنة قيمة معرف كل من الإجرائيتين بالأخرى Process Id.

4. التابع arrivalsort_MLFQ_RT_Time(const P_d& x, const P_d& y)

يُستخدم هذا التابع ضمن مراحل نمذجة خورزمية MLFQ-RT المُحسنة سابقاً في الورقة البحثية [1] وذلك لترتيب الإجرائيات بحسب زمن وصولها للنظام Arrival Time، حيث يأخذ التابع إجرائيتين للمقارنة بينهما من خلال مقارنة زمن وصول كل منهما بالآخر ويعيد القيمة البوليانية True في حال كان زمن وصول الإجرائية الأولى أصغر من ومن وصول الإجرائية الثانية إلى النظام. وعند تساوي أزمنة الوصول بين الإجرائيتين يتم مقارنة رشقة المعالج Burst Time لكل إجرائية مع الأخرى وعند التساوي أيضاً يتم مقارنة قيمة معرف كل من الإجرائيتين بالأخرى Process Id.

5. التابع arrivalsort_MLFQ_EDF_Time(const P_d& x, const P_d& y)

يُستخدم هذا التابع ضمن مراحل نمذجة الخوارزمية المُحسنة المقترحة في هذا البحث MLFQ-EDF لترتيب الإجرائيات بحسب زمن وصولها للنظام Arrival Time، حيث يأخذ التابع إجرائيتين للمقارنة بينهما من خلال مقارنة زمن وصول كل منهما بالآخر ويعيد القيمة البوليانية True في حال كان زمن وصول الإجرائية الأولى أصغر من ومن وصول الإجرائية الثانية إلى النظام.

عند تساوي أزمنة الوصول بين الإجرائيتين يتم مقارنة الموعد النهائي (الزمن الحرج Deadline) لكل إجرائية مع الأخرى وعند التساوي أيضاً يتم مقارنة قيمة معرف كل من الإجرائيتين بالأخرى.

التابع الرئيسي main():

تم استخدام بنية المعطيات vector لتخزين الإجرائيات من نمط الصف Process_Data والتي يتم توليدها يدوياً أو مولدة عشوائياً في النظام.

ويتم تمييز الحالتين التاليتين:

- إدخال الإجرائيات Processes بشكل يدوي

في هذه الحالة يتوجب على المستخدم إدخال عدد صحيح يمثل الإجرائيات n المطلوب جدولتها وفق خوارزمية الجدولة المتبعة، كما يتوجب عليه إدخال الخصائص التالية لكل إجرائية مُدخلة المُوضحة في الجدول (3-6).

الجدول (0-6): خصائص الإجرائيات Processes

خاصية الإجرائية	توضيح
Arrival Time (A_time)	زمن وصول الإجرائية إلى المُجدول ويكون عدداً صحيحاً
Burst Time (B_time)	زمن الإجرائية أو طول رشقة وحدة المعالجة المركزية المخصص للإجرائية
Deadline	متغير يمثل حالة كون الإجرائية المحددة المُدخلة هي إجرائية زمن حقيقي Real Time ويأخذ عدداً صحيحاً ويجب أن يحقق شرط

	أنه أكبر أو يساوي من مجموع زمن وصول الإجرائية مضافاً له طول الإجرائية (A_time + B_time) =< Deadline تم وضع رماز برمجي خاص للتحقق من الشرط السابق. في حال كانت Deadline المُدخلة من قبل المستخدم أكبر من 0 فهذا دليل على كون الإجرائية هي إجرائية زمن حقيقي RT وإذا كانت مساوية للصفر فهذا يدل أن الإجرائية هي عادية Non RT.

تم استخدام عدادين من النمط الصحيح لحساب عدد إجرائيات الزمن الحقيقي RT_counter والآخر لحساب عدد الإجرائيات العادية NRT_counter.

يتم اسناد قيمة طول الإجرائية B_time لقيمة الزمن المتبقي للإجرائية R_time وذلك قبل بداية تنفيذ خوارزمية الجدولة، كما يتم توليد معرف الإجرائية بشكل تلقائي متزايد Process Id في النظام.

- توليد الإجرائيات عشوائياً باستخدام Random Generator

في هذه الحالة نقوم بتوليد عينات الإجرائيات باستخدام مولد عينات عشوائي يقوم بتوليد مجموعة العينات المعيارية Standard Dataset وفق عدد صحيح n يتم إدخاله في البداية، بحيث يقوم المولد العشوائي بما يلي:

1. توليد معرف الإجرائية بشكل تلقائي متزايد لكل إجرائية (Process Id).
2. زمن وصول الإجرائية وهو عدد عشوائي صحيح يتراوح بين 0 و القيمة 100 (Arrival Time).
3. طول الإجرائية (زمن تنفيذها) وهو عدد عشوائي صحيح يتراوح بين 1 و 16 (Burst Time).
4. متغير بولياني للدلالة على إجرائية الزمن الحقيقي يأخذ قيمتين إما True للدلالة على أن الإجرائية الحالية هي إجرائية زمن حقيقي أو False للدلالة على أن الإجرائية ليست إجرائية زمن حقيقي (Is_RealTimeProcess).

5. الموعد النهائي لإجرائية الزمن الحقيقي (Deadline) وهو عدد عشوائي صحيح يتم حسابه من خلال جمع قيمة زمن وصول الإجرائية وطول الإجرائية (رشقتها) مضافاً له قيمة عشوائية تتراوح بين 1 و 100.

التابع ()sort:

يتم استخدام التابع sort لترتيب الإجرائيات ضمن vector بحسب زمن وصولها فالإجرائية الواصلة في زمن أقرب يتم إدخالها أولاً، بحيث تكون الإجرائيات مرتبة حسب زمن ورودها Arrival Time إلى النظام.

- حساب ومن التنفيذ الكلي للإجرائيات Total Execution Time (TET):

ويمثل عدد دورات الساعة الكلية clocks اللازمة لإنهاء جدولة جميع الإجرائيات الواردة إلى النظام، ويتم حسابه جمع قيم أطوال الإجرائيات Burst Time مع الأخذ بعين الاعتبار زمن وصول كل إجرائية إلى النظام.

ويتم تنجيز المُجدولات الثلاثة التي يتناولها هذا البحث بثلاثة توابع يظهرها الجدول التالي:

الجدول (0-7): التوابع المُمثلة لكل من خوارزميات الجدولة الثلاثة

ترتيب	التابع
1	MLFQ(input_MLFQ, n, total_exection_time, RT_counter)
2	MLFQ_RT(input_MLFQ_RT, n, total_exection_time,RT_counter)
3	MLFQ_EDF(input_MLFQ_EDF, n, total_exection_time, RT_counter)

في التابع رقم (1) تنجيز خوارزمية جدولة التغذية العكسية MLFQ الأساسية ويأخذ بارامتر مجموعة الإجرائيات المطلوب جدولتها input_MLFQ وبارامتر زمن التنفيذ الكلي total_execution_time وعدد إجرائيات الزمن الحقيقي RT_counter وسيتم شرح نمذجتها في فقرة خاصة لاحقة.

وفي التابع رقم (2) في تنجيز خوارزمية جدولة MLFQ-RT المقترحة سابقاً في الورقة البحثية [1] ويأخذ بارامتر مجموعة الإجرائيات المطلوب جدولتها input_MLFQ_RT وبارامتر زمن التنفيذ الكلي total_execution_time وعدد إجرائيات الزمن الحقيقي RT_counter وسيتم شرح نمذجتها في فقرة خاصة لاحقة.

بينما التابع رقم (3) يمثل تنجيز الخوارزمية المُقترحة في البحث MLFQ-EDF ويأخذ بارامتر مجموعة الإجرائيات المطلوب جدولتها input_MLFQ_EDF وبارامتر زمن التنفيذ الكلي total_execution_time وعدد إجرائيات الزمن الحقيقي RT_counter وسيتم شرح نمذجتها في فقرة خاصة لاحقة.

ثم يتم طباعة نتائج تقييم كل خوارزمية على الشاشة ليتم مقارنة نتائج الخوارزمية المُقترحة في البحث MLFQ_EDF مع الخوارزميتين MLFQ والخوارزمية MLFQ-RT وتحليل معايير الأداء.

3-1-1 تنجيز خوارزمية MLFQ

تم شرح طريقة تطبيق خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ في الفصل الثاني ولتحقيق ذلك عملياً تم استخدام مجموعة بنى المعطيات Data Structure والتوابع Functions اللازمة.

وخلال كل دورة ساعة (وحدة زمنية) clock من وقت المعالجة يتم تنفيذ خطوات الخوارزمية التي سبق ذكرها في شرح خطوات خوارزمية MLFQ في الفصل الثاني ويمكن تلخيصها وفق التالي:

1- إدخال الإجرائيات الواصلة حالياً إلى الرتل الأول الأعلى أولوية q1 وفق خوارزمية SRTF.
2- اختبار CPU فيما اذا كانت بحالة خمول أو حالة انشغال.
3- في حالة CPU خاملة يتم اختيار الإجرائية المرشحة للتنفيذ بناءاً على أولوية الرتل التي تنتمي الإجرائية له، فالرتل q1 المُجدول وفق خوارزمية SRTF له الأولوية الأعلى، وعندما يكون فارغاً يتم الانتقال لاختبار وجود الإجرائيات في الأرتال المتبقية بالترتيب لترشيحها للتنفيذ ضمن CPU.
4- في حالة CPU عاملة يتم تحديد مصدر الإجرائية الحالية Current ومن ثم مقارنة زمنها المتبقي R_time مع الأزمنة المتبقية للإجرائيات في الرتل الأول q1 وانتخاب الإجرائية ذات الزمن المتبقي الأصغر للتنفيذ وإخراج الإجرائية الحالية من CPU وإدخالها في الرتل q2. وهكذا يتم اختبار وجود

الإجرائيات في الأرتال جميعها مع الإجرائية الحالية وانتخاب الإجرائية المناسبة بناءاً عى أولويات الأرتال التي تنتمي لها الإجرائيات وعلى تسلسل وجود تلك الإجرائيات في أرتال الجهوزية Queues.

5- تنفيذ الإجرائية الحالية وإنقاص قيمة الزمن المتبقي لها R_time بمقدار 1 وإذا كان مصدر الإجرائية من أحد الأرتال الأربعة المُجدولة وفق خوارزمية الجدولة الدائرية Round Robin فيتم إنقاص قيمة الشريحة الزمنية المخصصة tq بمقدار 1.

6- اختبار حالة انتهاء تنفيذ الإجرائية الحالية في CPU من خلال التحقق من قيمة الزمن المتبقي للإجرائية R_time ويجب أن يكون مساوياً لـ 0 وفي هذة المرحلة يتم اتحقق من مصدر الإجرائية وتُصبح حالة CPU خاملة عند انتهاء تنفيذ الإجرائية وتغيير قيمة مؤشر الدلالة على مصدر الإجرائية الحالية أما بالنسبة للإجرائيات المُجدولة وفق RR فيتم اختبار قيمة الشريحة الزمنية لكل رتل من الأرتال وتصفير قيمة الشريجة الزمنية tq عند انتهاء الإجرائية الحالية أو مقاطعتها من قبل إجرائية أخرى أعلى أولوية High Priority.

7- ترتيب الإجرائيات في نهاية زمن التنفيذ الأعظمي tet وحساب معايير تقييم أداء خوارزمية MLFQ.

3-1-2 تنجيز خوارزمية MLFQ-RT

تم شرح طريقة تطبيق خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ-RT المُحسنة [1] في الفصل الثاني ولتحقيق ذلك عملياً نوضح الخطوات التالية:

تم استخدام مجموعة من أرتال الجهوزية عددها 5 أرتال لنمذجة خوارزمية MLFQ-RT وكل من هذه الأرتال تم جدولتا وفق خوارزمية جدولة معينة وفق الجدول (3-8).

الجدول (0-8): أرتال الجهوزية المستخدمة لخوارزمية MLFQ-RT

الرتل الأول الأعلى أولوية q1	تم استخدام خوارزمية الزمن الأقصر أولاً SJF لجدولة الإجرائيات في الرتل الأول
الرتل q2	تم استخدام خوارزمية Round Robin (RR) بشريحة زمنية tq2 = 2 لجدولة الإجرائيات في الرتل الثاني
الرتل q3	تم استخدام خوارزمية Round Robin (RR) بشريحة زمنية tq3 = 4 لجدولة الإجرائيات في الرتل الثالث
الرتل q4	تم استخدام خوارزمية Round Robin (RR) بشريحة زمنية tq4 = 8 لجدولة الإجرائيات في الرتل الرابع
الرتل q5	تم استخدام خوارزمية Round Robin (RR) بشريحة زمنية tq5 = 16 لجدولة الإجرائيات في الرتل الخامس

اقتراحَ المؤلف في الورقة البحثية [1] ما يلي لبناء خوارزمية MLFQ-RT:

- استخدام خوارزمية الزمن الأقصر أولاً Shortest Job First (SJF) في الرتل الأول q1 الأعلى أولوية.
- منع إجرائيات الرتل الأول q1 ذات أولوية الأعلى High Priority من التحرك ضمن باقي أرتال الجهوزية Ready Queues في المجدول.

يتم كتابة خطوات خوارزمية الجدولة MLFQ-RT وفق التالي:

1- إدخال الإجرائيات الواردة حالياً إلى المُجدول ضمن الرتل الأول الأعلى أولوية q1 المُجدول وفق خوارزمية Shortest Job First (SJF)

2- منع إجرائيات الرتل الأول q1 من التحرك ضمن باقي الأرتال عند حدوث مقاطعة لأي إجرائية مصدرها الرتل q1 بوجود إجرائية أخرى لها رشقة معالج Burst time أقل بحيث يُعاد إدخال الإجرائية الحالية في ذيل الرتل q1، مما يحقق اقتراح المؤلف في الورقة البحثية [1].

3- اختبار حالة CPU فيما اذا كانت خاملة أم بحالة انشغال.

4- في حالة CPU خاملة يتم اختيار الإجرائية المرشحة للتنفيذ بناءاً على أولوية الرتل التي تنتمي الإجرائية له، فالرتل q1 المُجدول وفق خوارزمية SJF له الأولوية الأعلى، وعندما يكون فارغاً يتم الانتقال لاختبار وجود الإجرائيات في الأرتال المتبقية بالترتيب لترشيحها للتنفيذ ضمن CPU.

5- أما اذا كانت CPU بحالة انشغال يتم تحديد مصدر الإجرائية الحالية Current ومن ثم مقارنة طول رشقة الإجرائية الحالية أو زمن تنفيذها Burst time (B_time) مع أزمنة التنفيذ للإجرائيات في الرتل الأول q1 وانتخاب الإجرائية ذات طول رشقة المعالج الأقل للتنفيذ وإخراج الإجرائية الحالية من CPU إعادة إدخالها في ذيل الرتل الأول q1.

6- التحقق من وجود الإجرائيات في باقي الأرتال q2 أو q3 أو q4 أو q5 وانتخاب الإجرائية المناسبة بناءاً عى أولويات الأرتال التي تنتمي لها الإجرائيات بحيث كل رتل أعلى أولوية يمنح إجرائياته أولوية أعلى عن الأرتال التي أدنى منه مما يجعل إي إجرائية منتظرة في أي رتل جهوزية Ready queue قادرة على مقاطعة إجرائية الرتل الأدنى والاستحواذ على CPU.

7- تنفيذ الإجرائية الحالية وإنقاص قيمة الزمن المتبقي لها R_time بمقدار 1 وإذا كان مصدر الإجرائية من أحد الأرتال الأربعة المُجدولة وفق خوارزمية RR فيتم إنقاص قيمة الشريحة الزمنية المخصصة tq بمقدار 1.

8- اختبار حالة انتهاء تنفيذ الإجرائية الحالية في CPU من خلال التحقق من قيمة الزمن المتبقي للإجرائية R_time ويجب أن يكون مساوياً لـ 0 وفي هذة المرحلة يتم التحقق من مصدر الإجرائية وتصبح CPU بحالة خمول عند انتهاء تنفيذ الإجرائية وتغيير قيمة مؤشر الدلالة على مصدر الإجرائية الحالية.

9- اختبار قيمة الشريحة الزمنية لكل رتل من أرتال الجهوزية بالنسبة للإجرائيات المُجدولة وفق RR فيتم وتصفير قيمة الشريجة الزمنية tq عند انتهاء الإجرائية الحالية أو مقاطعتها من قبل إجرائية أخرى أعلى أولوية High Priority.

10- ترتيب الإجرائيات في نهاية زمن التنفيذ الأعظمي Total Execution Time (TET) وحساب معايير أداء خوارزمية MLFQ-RT.

3-1-3 تنجيز الخوارزمية المُقترحة MLFQ-EDF

تم شرح طريقة تطبيق خوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ-EDF في الفصل الثاني ولتحقيق ذلك عملياً تم توضيح آلية الخوارزمية المُحسنة المقترحة المطبقة ومجموعة الخطوات وبنى المعطيات والتوابع اللازمة وفق مايلي:

وتقوم الخوارزمية المُقترحة بما يلي:

- استخدام خوارزمية الزمن النهائي الأقرب أولاً Earliest Deadline First (EDF) في الرتل الأول الأعلى أولوية q1 لجدولة إجرائيات الزمن الحقيقي RT بحيث تُدخل إجرائيات RT فور وصولها للنظام ضمن رتل الجهوزية الأول q1، وإعادة إدخال الإجرائيات التي تتم مقاطعتها بسبب إجرائيات أخرى واردة إلى إلى نهاية الرتل q1. مما يضمن أعلى نسبة نجاح لهذه الإجرائيات Success Ratio بمقابلتها للأزمنة النهائية Deadlines.
- منع إجرائيات الزمن الحقيقي RT الموجودة ضمن الرتل الأول q1 من الانتقال إلى باقي الأرتال مما يحقق أولوية عالية لهذه الإجرائيات عن باقي الإجرائيات العادية الأخرى.
- إدخال الإجرائيات العادية Non RT إلى رتل الجهوزية الثاني q2 المُجدول وفق خوارزمية Shortest Remining Time First (SRTF) مع إمكانية تنقيل الإجرائيات غير المنتهية إلى بقية الأرتال الأربعة المتبقية في المُجدول والتي يتم جدولتها وفق خوارزمية Round Robin (RR) بشرائح زمنية مختلفة.

يُقسم رتل الجهوزية إلى 6 أرتال تم تمثيلها ببنى المعطيات الموضّحة في الجدول (3-9).

الجدول (0-9): بنى المعطيات الممثلة لأرتال الجهوزية *Ready queues*

رتل الجهوزية Ready queue	الشرح
priority_queue (pq_rt)	الرتل الأول الأعلى أولوية q1 المُخصص لإجرائيات الزمن الحقيقي Real Time Processes وفيه يتم إدخال إجرائيات RT فقط وترتيبها ضمن الرتل بحسب الزمن الحرج الأقل للإجرائيات Deadline، واستخدام خوارزمية الزمن النهائي الأقرب أولاً EDF لجدولة الإجرائيات ضمن هذا الرتل مع منع إجرائيات الرتل الأول q1 من التنقيل لباقي الأرتال في حال مقاطعة تنفيذ أي منها.
priority_queue (pq)	إدخال الإجرائيات العادية Non RT في رتل الجهوزية الثاني q2 المُجدول وفق خوارزمية SRTF بحيث يمكن تنقيل الإجرائيات غير المنتهية ضمن باقي أرتال المجدول ذات الأولوية الأدنى.
queue rq3	رتل الجهوزية الثالث q3 المُجدول وفق خوارزمية Round Robin بشريحة زمنية tq3 = 2
queue rq4	رتل الجهوزية الرابع q4 المُجدول وفق خوارزمية Round Robin بشريحة زمنية tq4 = 4
queue rq5	رتل الجهوزية الخامس q5 المُجدول وفق خوارزمية Round Robin بشريحة زمنية tq5 = 8
queue rq6	رتل الجهوزية السادس q6 المُجدول وفق خوارزمية Round Robin بشريحة زمنية tq6 = 16

وتشمل خوارزمية الجدولة المُقترحة MLFQ-EDF الخطوات التالية:

1. إدخال إجرائيات الزمن الحقيقي RT ضمن الرتل الأول الأعلى أولوية q1 المخصص فقط لهذا النوع من الإجرائيات ويتم ترتيبها وفق الزمن النهائي الأقرب Deadline. تتم جدولة الإجرائيات وفق خوارزمية

جدولة الزمن النهائي الأقرب أولاً EDF مع منع إجرائيات الرتل q1 من الانتقال إلى أرتال المُجدول الأخرى.

2. إدخال الإجرائيات العادية Non RT ضمن الرتل الثاني q2 وتطبيق خوارزمية SRTF لجدولة الإجرائيات ضمن هذا الرتل مع إمكانية تنقيل الإجرائيات غير المنتهية أو التي تم مقاطعتها نزولاً ضمن باقي أرتال المُجدول.
3. اختبار حالة المعالج.

<u>في حال كان خاملاً Idle يتم تنفيذ الإجرائيات وفق التسلسل الآتي:</u>

- في حال كان الرتل q1 غير فارغ، يتم جدولة الإجرائية الأعلى أولوية وإسنادها للمعالج لتصبح حالته busy.
- في حال كان الرتل q1 فارغاً وكان الرتل q2 غير فارغ، تتم جدولة الإجرائية الأعلى أولوية وإسنادها للمعالج لتصبح حالته busy.
- في حال كان الرتل q1 والرتل q2 فارغين وكان الرتل q3 غير فارغ، تتم جدولة الإجرائية في مقدمة الرتل q3 لتتنفذ على المعالج زمن لايتجاوز الشريحة tq3 = 2 وتصبح حالة المعالج busy.
- في حال كان الرتل q1 والرتل q2 والرتل q3 فارغين وكان الرتل q4 غير فارغ، تتم جدولة الإجرائية في مقدمة الرتل q4 لتتنفذ على المعالج زمن لايتجاوز الشريحة tq4 = 4 وتصبح حالة المعالج busy.
- في حال كان الرتل q1 والرتل q2 والرتل q3 والرتل q4 فارغين وكان الرتل q5 غير فارغ، تتم جدولة الإجرائية في مقدمة الرتل q5 لتتنفذ على المعالج زمن لايتجاوز الشريحة tq5 = 8 وتصبح حالة المعالج busy.
- أما في حال كان الرتل q1 والرتل q2 والرتل q3 والرتل q4 والرتل q5 فارغين وكان الرتل q6 غير فارغ، تتم جدولة الإجرائية في مقدمة الرتل q6 لتتنفذ على المعالج زمن لايتجاوز الشريحة tq6 = 16 وتصبح حالة المعالج busy.

أما في حال كان المعالج مشغولاً Busy يتم تنفيذ الإجرائيات وفق التسلسل التالي:

- اذا كانت الإجرائية قيد التنفيذ قد جُدولت من الرتل q1 أي أن rt_process = 1 وكان هذا الرتل غير فارغ عندئذ تتم مقارنة الزمن النهائي لهذه الإجرائية مع الأزمنة النهائية للإجرائيات الموجودة في الرتل q1 والإجرائية ذات الزمن النهائي الأقرب Deadline هي التي تتابع التنفيذ على المعالج.
- اذا كانت الإجرائية الجارية التنفيذ قد جُدولت من الرتل q2 أي pq_process = 1 وكان الرتل q1 غير فارغ، سيتم الاستيلاء على هذه الإجرائية من قبل الإجرائية الأعلى أولوية بين إجرائيات الرتل q1 أي ستحرر المعالج وتنتقل إلى نهاية الرتل q3 وفي حال كان الرتل q1 فارغاً. يتم الاستيلاء على الإجرائية جارية التنفيذ على المعالج فقط في حال وجود إجرائية أخرى في الرتل q2 ذات زمن متبقي أصغر وتتغير قيمة المؤشر pq_process = 1.
- اذا كانت الإجرائية الجارية التنفيذ قد جُدولت من الرتل q3 أي rq3_process = 1 وكان الرتل q1 غير فارغ أو الرتل q2 غير فارغ. سيتم الاستيلاء على الإجرائية من قبل الإجرائية الأعلى أولوية من بين إجرائيات الرتلين q1 أو q2 أي ستحرر المعالج وتنتقل إلى نهاية الرتل q4 وتصبح قيمة المؤشر rt_process = 1 أو pq_process = 1 وذلك حسب رتل الإجرائية وتصبح حالة مؤشر الرتل الثالث rq3_process = 0.
- اذا كانت الإجرائية الجارية التنفيذ قد جُدولت من الرتل q4 أي rq4_process = 1 وكان أحد الأرتال q1 أو q2 أو q3 غير فارغ فيتم الاستيلاء على الإجرائية الحالية من قبل الإجرائية الأعلى أولوية من بين إجرائيات الأرتال q1 أو q2 أو q3 أي ستحرر المعالج وتنتقل إلى نهاية الرتل q5 ويصبح مؤشر رتل الإجرائية الجديدة في المعالج 1 وتصبح قيمة المؤشر rq4_process = 0.
- اذا كانت الإجرائية جارية التنفيذ في المعالج من الرتل q5 أي rq5_process = 1 وإحدى الأرتال (q1…q4) غير فارغ، فيتم الاستيلاء على الإجرائية الحالية من قبل الإجرائية الأعلى أولوية High Priority من بين إجرائيات الأرتال الأربعة السابقة أي ستحرر المعالج وتنتقل

إلى نهاية الرتل السادس q6 ويصبح مؤشر رتل الإجرائية الجديدة في المعالج 1 وتصبح قيمة المؤشر rq5_process = 0.

- اذا كانت الإجرائية جارية التنفيذ في المعالج من الرتل q6 أي rq6_process = 1 وإحدى الأرتال (q1…q5) غير فارغ، فيتم الاستيلاء على الإجرائية الحالية من قبل الإجرائية الأعلى أولوية High Priority من بين إجرائيات الأرتال الخمسة السابقة أي ستحرر المعالج ويُعاد إدخالها إلى نهاية الرتل السادس q6 ويصبح مؤشر رتل الإجرائية الجديدة في المعالج 1 وتصبح قيمة المؤشر rq6_process = 0.

4. تنفيذ الإجرائية الحالية Current Process وإنقاص قيمة الزمن المتبقي R_time بالمقدار 1 واذا كان الإجرائية الحالية من أحد الأرتال الأربعة q3 أو q4 أو q5 أو q6 فيتم إنقاص قيمة الشريحة الزمنية المخصصة quantum بمقدار 1.
5. اختبار حالة انتهاء أو اكتمال تنفيذ الإجرائية الحالية وفق (R_time = 0) ونميز هنا الحالتين التاليتين:

حالة انتهاء تنفيذ الإجرائية الحالية

- انتهاء تنفيذ إجرائية من الرتل q1 يتم تحويل حالة CPU إلى خمول Idle واسناد قيمة rt_process = 0.
- انتهاء تنفيذ إجرائية من الرتل q2 يتم تحويل حالة CPU إلى خمول واسناد قيمة pq_process = 0.
- انتهاء تنفيذ إجرائية من الرتل q3 يتم تحويل حالة CPU إلى خمول Idle واسناد قيمة rq3_process = 0 وتهيئة قيمة الشريحة الزمنية tq3 = 2 للإجرائية الجديدة الواردة.
- انتهاء تنفيذ إجرائية من الرتل q4 يتم تحويل حالة CPU إلى خمول Idle واسناد قيمة rq4_process = 0 وتهيئة قيمة الشريحة الزمنية tq4 = 4 للإجرائية الجديدة الواردة.
- انتهاء تنفيذ إجرائية من الرتل q5 يتم تحويل حالة CPU إلى خمول Idle واسناد قيمة rq5_process = 0 وتهيئة قيمة الشريحة الزمنية tq5 = 8 للإجرائية الجديدة الواردة.
- انتهاء تنفيذ إجرائية من الرتل q6 يتم تحويل حالة CPU إلى خمول Idle واسناد قيمة rq6_process = 0 وتهيئة قيمة الشريحة الزمنية tq6 = 16 للإجرائية الجديدة الواردة.

حالة عدم انتهاء تنفيذ الإجرائية الحالية

- اذا كانت الإجرائية الحالية من الرتل q3 وانتهت قيمة الشريحة الزمنية المخصصة لهذه الإجرائية بحيث أصبحت (tq3 = 0)، فيتم الاستيلاء على الإجرائية الحالية وتنتقل إلى ذيل الرتل q4 وتحويل حالة CPU إلى خمول Idle وحالة المؤشر (rq3_process = 0).
- اذا كانت الإجرائية الحالية من الرتل q4 وانتهت قيمة الشريحة الزمنية المخصصة لهذه الإجرائية بحيث أصبحت (tq4 = 0)، فيتم الاستيلاء على الإجرائية الحالية وتنتقل إلى ذيل الرتل q5 وتحويل حالة CPU إلى Idle وحالة المؤشر (rq4_process = 0).
- اذا كانت الإجرائية الحالية من الرتل q5 وانتهت قيمة الشريحة الزمنية المخصصة لهذه الإجرائية بحيث أصبحت (tq5 = 0)، الاستيلاء على الإجرائية الحالية وتنتقل إلى ذيل الرتل q6 وتحويل حالة CPU إلى خمول Idle وحالة المؤشر (rq5_process = 0).
- اذا كانت الإجرائية الحالية من الرتل q6 وانتهت قيمة الشريحة الزمنية المخصصة لهذه الإجرائية بحيث أصبحت (tq6 = 0)، فيتم الاستيلاء على الإجرائية الحالية ويُعاد إدخالها في ذيل الرتل q6 وتحويل حالة CPU إلى خمول وحالة مؤشر دلالة مصدر الإجرائية للقيمة (rq5_process = 0).

6. ترتيب الإجرائيات بحسب معرف الإجرائية Process Id وحساب زمن انتهاء (اكتمال) Finish Time كل إجرائية مُدخلة إلى المجدول وزمن بدء تنفيذ كل إجرائية Start Time أيضاً. كما يتم حساب زمن الانتظار Waiting Time (W_time) وزمن التنفيذ الكلي Turnaround Time(T_time) لكل إجرائية.
7. حساب معايير تقييم أداء الخوارزمية المُحسنة المقترحة وتشمل زمن الانتظار الوسطي و زمن التنفيذ الوسطي ومعدل نجاح إجرائيات الزمن الحقيقي RT.
8. عرض النتائج ومقارنتها مع الخوارزميات السابقة.

3-2 النتائج العملية.

يُراد تجريب الخوارزميات الثلاثة التي تم دراستها وشرحها في الفصول السابقة وتسجيل النتائج وتحليلها حيث يتم توليد مجموعات معيارية مختلفة من الإجرائيات عشوائياً باستخدام مولد معياري عشوائي للعينات Random Generator وفق منهجية توليد العينات المعيارية التي سبق وتم توضيحها في فقرة التطبيق العملي يقوم بتوليد العينات وخصائها لتكون قابلة للتجريب على بيئة التطوير والتنفيذ المستخدمة في البحث.

سيتم تطبيق اختبارين على مجموعات مختلفة وكبيرة من الإجرائيات العشوائية وتطبيق المجموعات على كل من الخوارزميات الثلاثة قيد الدراسة.

يتم تحديد دورة الساعة clock بالميلي ثانية (1 ms) عند تنفيذ الإجرائيات ضمن المجدولات المطبقة في التنجيز. وتتم دراسة النتائج وفق ثلاثة معايير أساسية وهم:

- معدل نجاح إجرائيات الزمن الحقيقي Success Ratio
- متوسط أزمنة انتظار الإجرائيات Average Waiting Time
- متوسط أزمنة التنفيذ الكلية الإجرائيات Average Turnaround Time

تسعى الخوارزمية المُقترحة في البحث للحصول على أعلى معدل نجاح لإجرائيات الزمن الحقيقي Success Ratio والحصول على زمن أصغري لكل من متوسط أزمنة انتظار الإجرائيات Average Waiting Time ومتوسط أزمنة تنفيذ الإجرائيات Average Turnaround Time.

3-2-1 الاختبار الأول (المجموعة الأولى)

في هذا الاختبار يتم اختبار الخوارزميات الثلاثة على مجموعة إجرائيات مولدة عشوائياً ويتراوح عدد الإجرائيات بين المجال **100** و **1000**.

يتم بعد ذلك مقارنة النتائج التي تعطيها كل خوارزمية مع كل من خوارزمية MLFQ وخوارزمية MLFQ-RT والخوارزمية المُقترحة MLFQ-EDF.

ويوضح الشكل (3-1) الإجرائيات المولدة عشوائياً التي تم التجريب عليها.

>>>>>>>>>>>>>>>>>>>   input : <<<<<<<<<<<<<<<<<

Pid	A_time	B_time	Is_Real_Time	Deadline
1	56	10	true	155
2	79	6	false	0
3	95	1	false	0
4	76	11	true	119
5	10	6	true	74
6	6	13	false	0
7	2	2	true	9
8	34	5	true	52
9	8	16	false	0
10	30	5	false	0
11	61	7	false	0
12	18	8	true	35
13	40	16	true	70
14	83	16	false	0
15	8	2	true	64
16	96	2	true	192
17	16	16	false	0
18	24	15	true	66
19	18	16	true	54
20	88	6	false	0
21	84	3	true	127
22	80	7	false	0
23	77	12	true	176
24	91	10	true	112
25	57	1	false	0
26	94	16	false	0
27	59	6	true	139
28	39	1	false	0
29	35	10	true	111
30	29	6	true	48
31	46	11	true	119
32	81	10	false	0
33	16	14	false	0
34	23	2	true	77
35	99	2	true	195
36	30	2	true	91
37	73	9	true	95
38	68	7	false	0
39	66	8	true	153
40	71	8	true	172

⋮

الشكل (3-1): الإجرائيات التي تم التجريب عليها في الاختبار الأول

نتائج تنفيذ خوارزمية MLFQ

ويظهر الجدول (3-10) نتائج الاختبار الأول باستخدام الخوارزمية MLFQ.

الجدول (0-10): نتائج الاختبار الأول على خوارزمية MLFQ

MLFQ Algorithm			
Processes (count)	Success Ratio	Avg Waiting Time (ms)	Avg Turnaround Time (ms)
100	0.07	501	509
150	0.07	775	784
200	0.03	1092	1101
250	0.008	1334	1343
300	0.02	1575	1584
350	0.02	1950	1956
400	0.016	2153	2161
450	0.008	2435	2443
500	0.013	2626	2634
550	0.01	3013	3022
600	0.025	3238	3246
650	0.009	3610	3619
700	0.005	3866	3874
750	0.013	4114	4123
800	0.012	4391	4400
850	0.009	4673	4681
900	0.015	5025	5033
950	0.008	5199	5207
1000	0.006	5369	5377

بملاحظة النتائج السابقة نجد أن خوارزمية MLFQ أعطت أفضل نتيجة لنسبة نجاح إجرائيات الزمن الحقيقي Success Ratio على جميع مجموعات الإجرائيات وهو 0.07 وأصغر متوسط أزمنة انتظار Avg Waiting Time وهو 501، وأصغر متوسط أزمنة تنفيذ كُلية Avg Turnaround Time وهو 509.

نتائج تنفيذ خوارزمية MLFQ-RT المقترحة في الورقة البحثية [1].

ويظهر الجدول (3-11) نتائج الاختبار الأول باستخدام الخوارزمية MLFQ-RT.

الجدول (0-11): نتائج الاختبار الأول على خوارزمية MLFQ-RT

MLFQ-RT Algorithm			
Processes (count)	Success Ratio	Avg Waiting Time (ms)	Avg Turnaround Time (ms)
100	0.117	420	428
150	0.09	682	690
200	0.078	948	956
250	0.06	1161	1170
300	0.05	1362	1370
350	0.03	1691	1700
400	0.043	2015	1973
450	0.05	2054	2063
500	0.033	2398	2353
550	0.018	2574	2582
600	0.028	2836	2845
650	0.018	3260	3153
700	0.017	3527	3335
750	0.021	3585	3593
800	0.019	3852	3861
850	0.015	4201	4014
900	0.022	4712	4361
950	0.014	4564	4572
1000	0.018	5012	4906

بملاحظة النتائج السابقة نجد أن خوارزمية MLFQ-RT أعطت أفضل نتيجة لنسبة نجاح إجرائيات الزمن الحقيقي Success Ratio عند 0.25، وأصغر متوسط أزمنة انتظار Avg Waiting Time على مجوعات الإجرتئيات وهو 420، وأصغر متوسط أزمنة تنفيذ كلية Avg Turnaround Time للإجرائيات وهو 428.

نلاحظ أن الحل الذي توصلت إليه خوارزمية MLFQ-RT أفضل من الحل الذي توصلت إليه خوارزمية MLFQ من أجل نفس مجموعات الإجرائيات في الاختبار الأول.

نتائج تنفيذ الخوارزمية المُقترحة MLFQ-EDF

ويظهر الجدول (3-12) نتائج الاختبار الأول باستخدام الخوارزمية MLFQ-EDF.

الجدول (0-12): نتائج الاختبار الأول على الخوارزمية المقترحة MLFQ-EDF

MLFQ-EDF Algorithm			
Processes (count)	Success Ratio	Avg Waiting Time (ms)	Avg Turnaround Time (ms)
100	0.32	334	341
150	0.16	664	672
200	0.156	837	845
250	0.15	1103	1112
300	0.11	1310	1315
350	0.1	1575	1370
400	0.076	1782	1790
450	0.08	1960	1971
500	0.057	2305	2314
550	0.06	2475	2484
600	0.051	2645	2653
650	0.052	2993	3001
700	0.047	3148	3157
750	0.043	3392	3401
800	0.039	3605	3518
850	0.036	3793	3801
900	0.039	4125	4134
950	0.038	4312	4320
1000	0.029	4560	4569

بملاحظة النتائج السابقة نجد أن الخوارزمية المُحسنة المقترحة في البحث MLFQ-EDF أعطت أفضل نتيجة لنسبة نجاح إجرائيات الزمن الحقيقي Success Ratio عند 0.35، وأصغر متوسط أزمنة انتظار Avg Waiting Time على مجوعات الإجرائيات المختبرة وهو 334، وأصغر متوسط أزمنة تنفيذ كلية Avg Turnaround Time وهو 341.

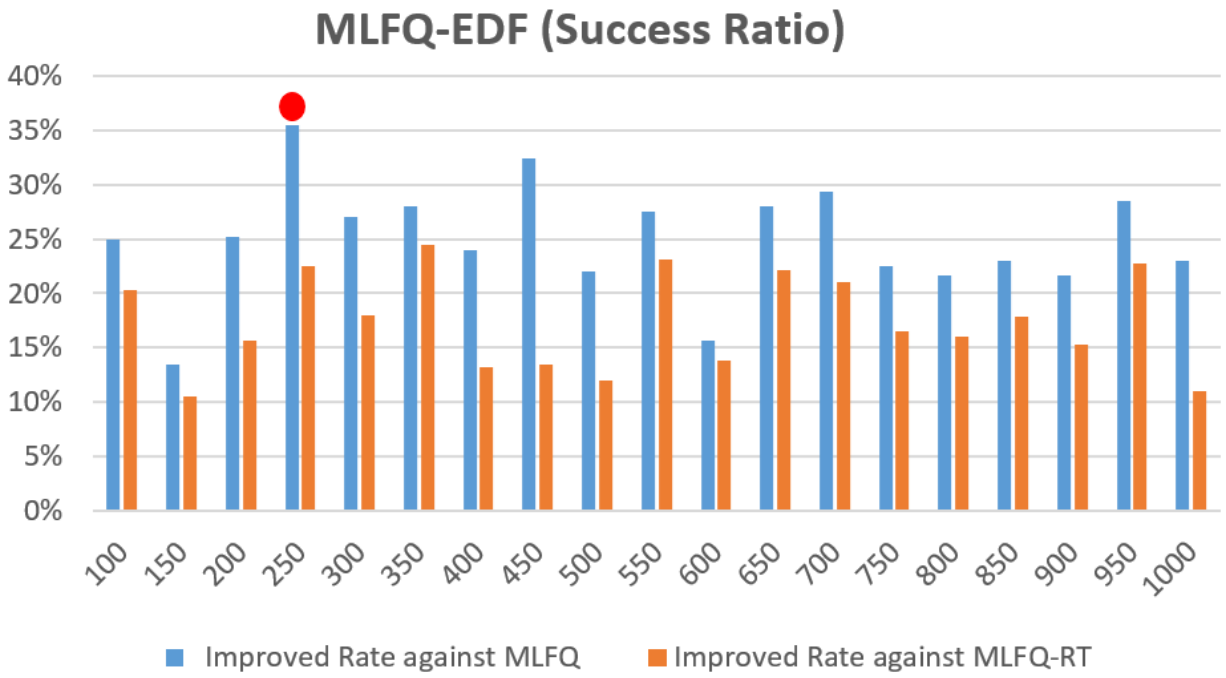
نلاحظ من نتائج تطبيق الاختبارات على الخوارزميات الثلاثة أن الخوارزمية المقترحة MLFQ-EDF أعطت أفضل النتائج بين الخوارزميات الثلاثة وفق معايير تقييم الأداء.

بمقارنة نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio في الخوارزمية المقترحة MLFQ-EDF مع كل من الخوارزميتين MLFQ و MLFQ-RT في الجدول (3-13):

الجدول (0-13): مقارنة نسبة النجاح Success Ratio (الاختبار الأول) بين الخوارزميات الثلاثة.

Co. of Processes	Success Ratio Improved Algorithm (MLFQ-EDF)	Success Ratio (MLFQ-RT)	Success Ratio (MLFQ)	Improved Rate (MLFQ-RT) %	Improved Rate (MLFQ) %
100	0.32	0.117	0.07	20.3%	25%
150	0.16	0.09	0.07	10.5%	13.5%
200	0.156	0.078	0.03	15.6%	25.2%
250	0.15	0.06	0.008	22.5%	35.5%
300	0.11	0.05	0.02	18%	27%
350	0.1	0.03	0.02	24.5%	28%
400	0.076	0.043	0.016	13.2%	24%
450	0.08	0.05	0.008	13.5%	32.4%
500	0.057	0.033	0.013	12%	22%
550	0.06	0.018	0.01	23.1%	27.5%
600	0.051	0.028	0.025	13.8%	15.6%
650	0.052	0.018	0.009	22.1%	27.95%
700	0.047	0.017	0.005	21%	29.4%
750	0.043	0.021	0.013	16.5%	22.5%
800	0.039	0.019	0.012	16%	21.6%
850	0.036	0.015	0.009	17.85%	22.95%
900	0.039	0.022	0.015	15.3%	21.6%
950	0.038	0.014	0.008	22.8%	28.5%
1000	0.029	0.018	0.006	11%	23%
Success Ratio Average :				**17.3%**	**25%**

نُلاحظ من تحليل نتائج الاختبار الأول أن الخوارزمية المُحسنة المقترحة MLFQ-EDF أعطت نتائج أفضل بالنسبة لنسبة نجاح إجرائيات الزمن الحقيقي Success Ratio، بنسبة 25% عن خوارزمية MLFQ وبنسبة 17.3% عن خوارزمية MLFQ-RT المقترحة في الورقة البحثية [1].



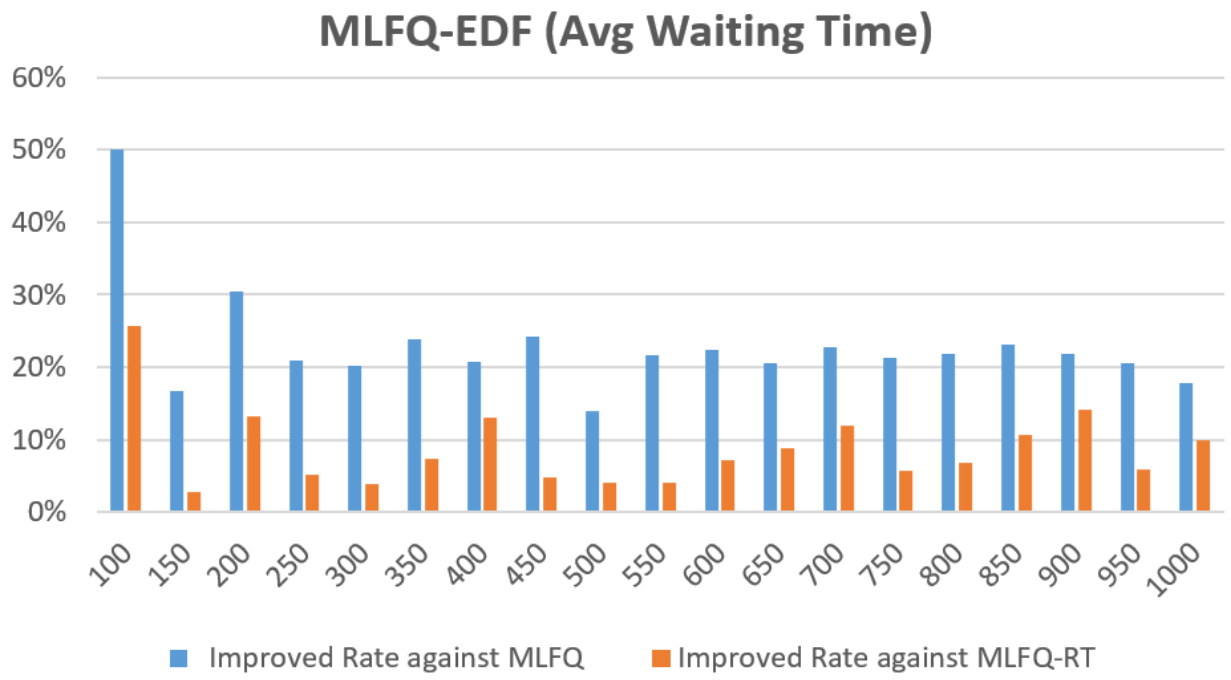
الشكل (3-2): تحسن نسبة نجاح إجرائيات الزمن الحقيقي (الاختبار الأول) MLFQ-EDF

يوضح الجدول (3-14) بمقارنة متوسط أزمنة انتظار الإجرائيات Avg Waiting Time في الخوارزمية المقترحة MLFQ-EDF مع كل من الخوارزميتين MLFQ و MLFQ-RT.

الجدول (0-14): مقارنة نسب تحسين متوسط أزمنة انتظار الإجرائيات (الاختبار الأول)

Co. of Processes	**Avg Waiting Time Improved Algorithm (MLFQ-EDF)** (ms)	**Avg Waiting Time (MLFQ-RT)** (ms)	**Avg Waiting Time (MLFQ)** (ms)	**Improved Rate (MLFQ-RT) %**	**Improved Rate (MLFQ) %**
100	334	420	501	25.74%	50%
150	664	682	775	2.71%	16.71%
200	837	948	1092	13.26%	30.46%
250	1103	1161	1334	5.25%	20.94%
300	1310	1362	1575	3.96%	20.22%
350	1575	1691	1950	7.36%	23.80%
400	1782	2015	2153	13.07%	20.81%
450	1960	2054	2435	4.79%	24.23%
500	2305	2398	2626	4.03%	13.92%
550	2475	2574	3013	4%	21.73%
600	2645	2836	3238	7.22%	22.41%
650	2993	3260	3610	8.92%	20.61%
700	3148	3527	3866	12.03%	22.80%
750	3392	3585	4114	5.68%	21.28%
800	3605	3852	4391	6.85%	21.80%
850	3793	4201	4673	10.75%	23.20%
900	4125	4712	5025	14.23%	21.81%
950	4312	4564	5199	5.84%	20.57%
1000	4560	5012	5369	9.91%	17.74%
Waiting Time Average :				**8.72%**	**23%**

نُلاحظ من تحليل نتائج الاختبار الاختبار أن الخوارزمية المُقترحة MLFQ-EDF حسنت من متوسط أزمنة انتظار الإجرائيات Avg Waiting Time بنسبة 23% عن خوارزمية MLFQ وبنسبة 8.72% عن خوارزمية MLFQ-RT. ويوضح الشكل (3-3) نسب التحسن.



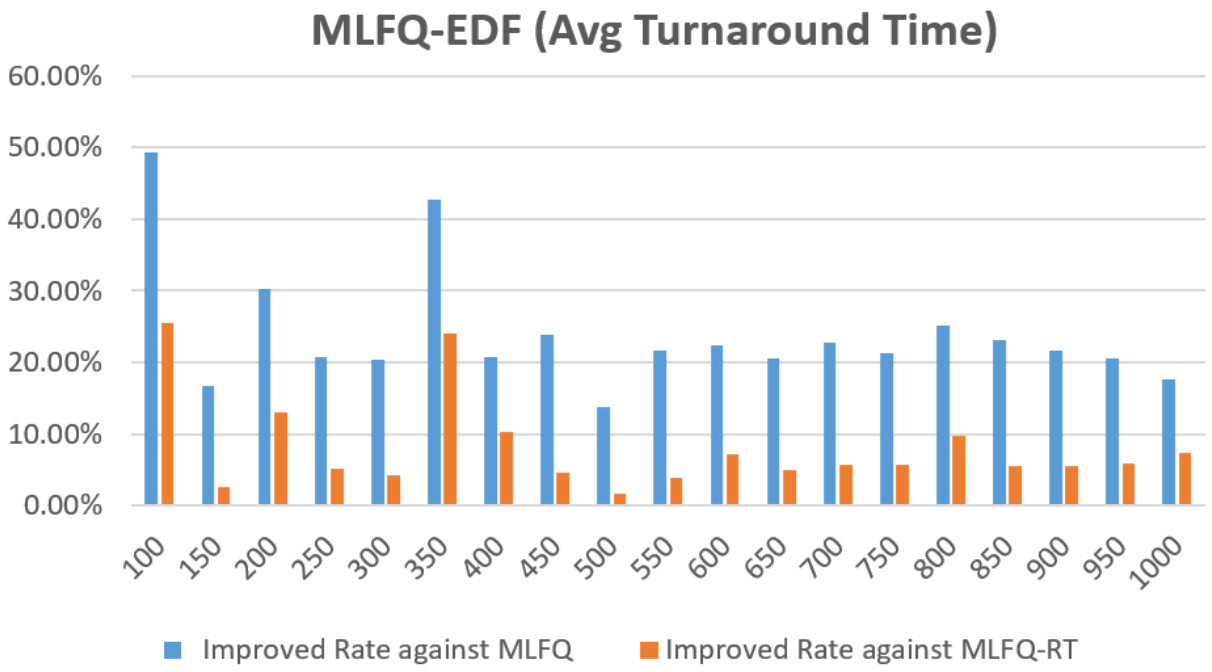
الشكل (3-3): تحسن متوسط أزمنة انتظار الإجرائيات (الاختبار الأول) MLFQ-EDF

ويوضح الجدول (3-15) مقارنة متوسط أزمنة التنفيذ الكلية للإجرائيات Avg Turnaround Time في الخوارزمية المقترحة MLFQ-EDF مع كل من الخوارزميتين MLFQ و MLFQ-RT.

الجدول (0-15): مقارنة نسب تحسين متوسط أزمنة التنفيذ الكلية للإجرائيات (الاختبار الأول).

Co. of Processes	**Avg Turnaround Time Improved Algorithm (MLFQ-EDF)** (ms)	**Avg Turnaround Time (MLFQ-RT)** (ms)	**Avg Turnaround Time (MLFQ)** (ms)	**Improved Rate (MLFQ-RT) %**	**Improved Rate (MLFQ) %**
100	341	428	509	25.51%	%49.26
150	672	690	784	%2.67	%16.66
200	845	956	1101	%13.13	%30.29
250	1112	1170	1343	%5.21	%20.77
300	1315	1370	1584	%4.18	%20.45
350	1370	1700	1956	%24.08	%42.77
400	1790	1973	2161	%10.22	%20.72
450	1971	2063	2443	%4.66	%23.94
500	2314	2353	2634	%1.68	%13.82
550	2484	2582	3022	%3.94	%21.65
600	2653	2845	3246	%7.23	%22.35
650	3001	3153	3619	%5.06	%20.59
700	3157	3335	3874	%5.63	%22.71
750	3401	3593	4123	%5.64	%21.22
800	3518	3861	4400	%9.74	%25.07
850	3801	4014	4681	%5.60	%23.15
900	4134	4361	5033	%5.49	%21.74
950	4320	4572	5207	5.83%	%20.53
1000	4569	4906	5377	%7.37	%17.68
Turnaround Time Average :				**8%**	**24%**

نُلاحظ من تحليل نتائج الاختبار الأول أن الخوارزمية المُقترحة MLFQ-EDF حَسنت من متوسط أزمنة التنفيذ الكلية للإجرائيات Avg Turnaround Time بنسبة 24% عن خوارزمية MLFQ وبنسبة 8% عن خوارزمية MLFQ-RT. ويوضح الشكل (3-4) نسب التحسن.



الشكل (3-4): تحسن معيار متوسط تنفيذ الإجرائيات (الاختبار الأول) في خوارزمية MLFQ-EDF

نسب التحسين للاختبار الأول على المجموعة الأولى:

يوضح الجدول (3-16) نسب تحسن الخوارزمية المقترحة MLFQ-EDF عن كل من الخوارزميتين MLFQ وMLFQ-RT وفق معايير تقييم الأداء للاختبار الأول.

الجدول (0-16): نسب التحسين في الاختبار الأول

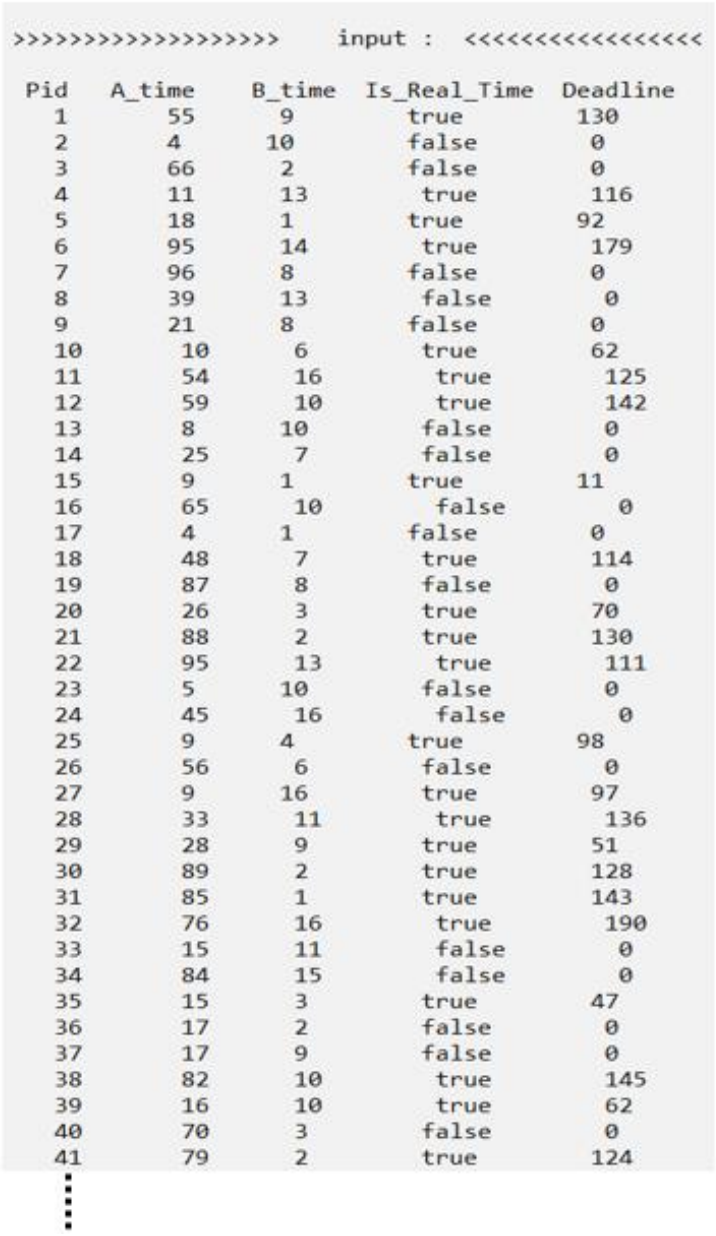
المعيار	معدل تَحسن خوارزمية MLFQ-EDF عن MLFQ-RT	معدل تَحسن خوارزمية MLFQ-EDF عن MLFQ
نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio	17.3%	25%
متوسط أزمنة انتظار الإجرائيات Avg Waiting Time	8.72%	23%
متوسط أزمنة التنفيذ الكلية للإجرائيات Avg Turnaround Time	8%	24%

3-2-2 الاختبار الثاني (المجموعة الثانية)

في هذا الاختبار يتم اختبار الخوارزميات الثلاثة على مجموعة إجرائيات مولدة عشوائياً ويتراوح عدد الإجرائيات بين المجال **1000** و **7000**.

يتم بعد ذلك مقارنة النتائج التي تعطيها كل خوارزمية مع كل من خوارزمية MLFQ وخوارزمية MLFQ-RT والخوارزمية المُقترحة MLFQ-EDF.

ويوضح الشكل (3-5) الإجرائيات المولدة عشوائياً التي تم التجريب عليها.



```
>>>>>>>>>>>>>>>>>>    input :  <<<<<<<<<<<<<<<<<

 Pid   A_time    B_time  Is_Real_Time  Deadline
   1       55       9         true        130
   2        4      10        false         0
   3       66       2        false         0
   4       11      13         true        116
   5       18       1         true        92
   6       95      14         true        179
   7       96       8        false         0
   8       39      13         false         0
   9       21       8        false         0
  10        10       6         true        62
  11        54      16          true        125
  12        59      10          true        142
  13        8       10         false         0
  14        25       7         false         0
  15        9        1         true        11
  16        65       10          false         0
  17        4        1         false         0
  18        48       7         true        114
  19        87       8         false         0
  20        26       3         true        70
  21        88       2         true        130
  22        95      13          true        111
  23        5       10         false         0
  24        45       16          false         0
  25        9        4         true        98
  26        56       6         false         0
  27        9       16         true        97
  28        33       11          true        136
  29        28       9         true        51
  30        89       2         true        128
  31        85       1         true        143
  32        76      16          true        190
  33        15      11          false         0
  34        84      15          false         0
  35        15       3         true        47
  36        17       2         false         0
  37        17       9         false         0
  38        82      10          true        145
  39        16      10          true        62
  40        70       3         false         0
  41        79       2         true        124
```

الشكل (3-5): الإجرائيات التي تم الاختبار عليها في الاختبار الثاني

<u>نتائج تنفيذ خوارزمية MLFQ</u>

ويوضح الجدول (3-17) نتائج الاختبار الثاني باستخدام الخوارزمية MLFQ.

الجدول (0-17): نتائج الاختبار الثاني على خوارزمية MLFQ

MLFQ Algorithm			
Processes (count)	Success Ratio	Avg Waiting Time (ms)	Avg Turnaround Time (ms)
1200	0.005	6740	6749
1500	0.007	8270	8278
2000	0.004	11012	11020
2400	0.002	13147	13156
2800	0.001	15658	15666
3000	0.003	16601	16610
3500	0.002	19541	19549
4000	0.002	22131	22140
4200	0.0009	23348	23357
4900	0.0012	27258	27267
5000	0.0016	27848	27857
5500	0.0018	30931	30939
6000	0.0016	33160	33168
6700	0.0005	37279	37287
7000	0.0002	39005	39014

بملاحظة النتائج السابقة نجد أن خوارزمية MLFQ حسّنت نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio على جميع الإجرائيات وهو 0.007 وأصغر متوسط أزمنة انتظار Avg Waiting Time على الإجرائيات المختبرة وهو 6740، وأصغر متوسط أزمنة تنفيذ كلية Avg Turnaround Time على الإجرائيات المختبرة وهو 6749.

<u>نتائج تنفيذ خوارزمية MLFQ-RT</u> المقترحة في الورقة البحثية [1]

يوضح الجدول (3-18) نتائج تنفيذ الاختبار الثاني باستخدام خوارزمية MLFQ-RT.

الجدول(3-18): نتائج الاختبار الثاني على خوارزمية MLFQ-RT

MLFQ-RT Algorithm			
Processes (count)	Success Ratio	Avg Waiting Time (ms)	Avg Turnaround Time (ms)
1200	0.01	5938	5947
1500	0.011	7243	7251
2000	0.008	9609	9855
2400	0.005	12540	11442
2800	0.004	13626	13635
3000	0.005	14388	14396
3500	0.005	17072	17081
4000	0.004	20150	19690
4200	0.003	20384	20393
4900	0.002	23844	23852
5000	0.0032	25890	24273
5500	0.0022	27042	27051
6000	0.0016	28884	28892
6700	0.0023	33120	32820
7000	0.0017	33938	33946

بملاحظة النتائج السابقة نجد أن خوارزمية MLFQ-RT حسَّنت نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio عند 0.011، وأصغر متوسط أزمنة انتظار Avg Waiting Time على مجموعات الإجرائيات المختبرة وهو 5938، وأصغر متوسط أزمنة تنفيذ كلية Avg Turnaround Time على الإجرائيات المختبرة وهو 5947.

نلاحظ أن الحل الذي توصلت إليه خوارزمية MLFQ-RT أفضل من الحل الذي توصلت إليه خوارزمية MLFQ من أجل نفس مجموعات الإجرائيات في الاختبار الأول.

نتائج تنفيذ الخوارزمية المُقترحة MLFQ-EDF

يوضح الجدول (3-19) نتائج تنفيذ الاختبار الثاني على خوارزمية MLFQ-EDF.

الجدول (0–18): نتائج الاختبار الثاني على الخوارزمية المقترحة MLFQ-EDF

MLFQ-EDF Algorithm			
Processes (count)	Success Ratio	Avg Waiting Time (ms)	Avg Turnaround Time (ms)
1200	0.025	5581	5590
1500	0.02	6839	6847
2000	0.015	9138	9147
2400	0.012	10880	10889
2800	0.0127	12845	12854
3000	0.01	13765	13773
3500	0.008	16220	16229
4000	0.007	18359	18367
4200	0.0073	19468	19477
4900	0.0057	22641	22650
5000	0.0064	22942	22951
5500	0.0051	25851	25860
6000	0.0046	27456	27464
6700	0.0041	30769	30778
7000	0.004	32265	32274

بملاحظة النتائج السابقة نجد أن الخوارزمية المُقترحة في البحث MLFQ-EDF أعطت أفضل نتيجة لنسبة نجاح إجرائيات الزمن الحقيقي Success Ratio عند 0.025، وأصغر متوسط أزمنة انتظار Avg Waiting Time على مجموعات الإجرائيات المختبرة وهو 5581، وأصغر متوسط أزمنة تنفيذ كلية Avg Turnaround Time على الإجرائيات المختبرة وهو 5590.

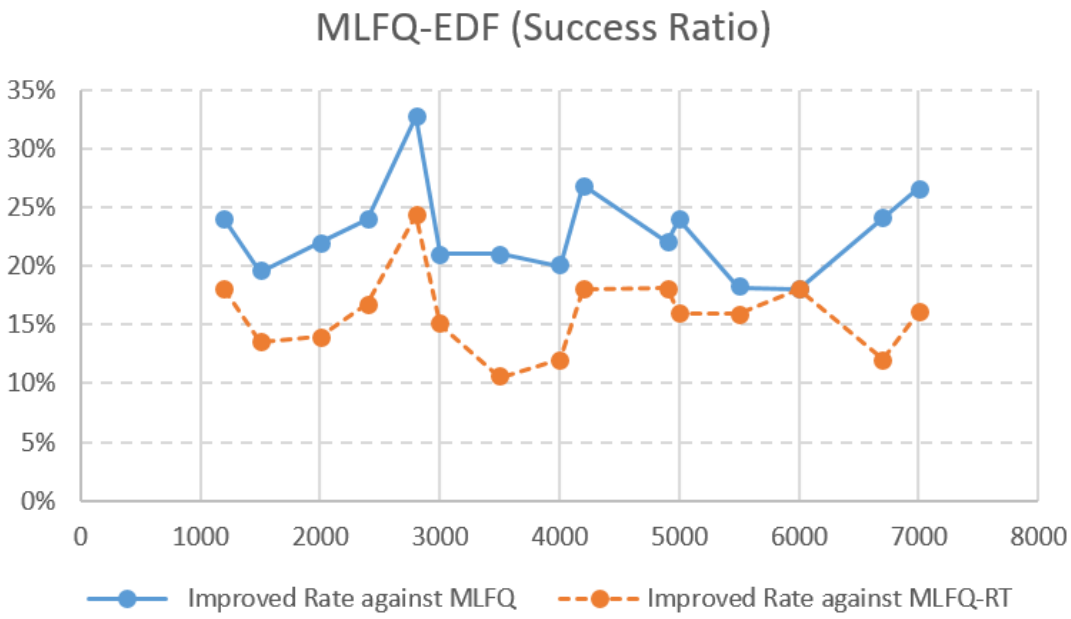
نلاحظ من نتائج تطبيق الاختبارات على الخوارزميات الثلاثة أن الخوارزمية المقترحة MLFQ-EDF تفوقت على خوارزميتي MLFQ وَ MLFQ-RT وفق معايير تقييم الأداء.

يوضح الجدول (3–20) مقارنة نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio في الخوارزمية المقترحة MLFQ-EDF مع كل من خوارزميتي MLFQ و MLFQ-RT

الجدول (0-19): مقارنة نسبة نجاح إجرائيات الزمن الحقيقي (الاختبار الثاني).

Co. of Processes	Success Ratio Improved Algorithm (MLFQ-EDF)	Success Ratio (MLFQ-RT)	Success Ratio (MLFQ)	Improved Rate (MLFQ-RT) %	Improved Rate (MLFQ) %
1200	0.025	0.01	0.005	18%	24%
1500	0.02	0.011	0.007	13.5%	19.5%
2000	0.015	0.008	0.004	14%	22%
2400	0.012	0.005	0.002	16.8%	24%
2800	0.0127	0.004	0.001	24.36%	32.76%
3000	0.01	0.005	0.003	15%	21%
3500	0.008	0.005	0.002	10.5%	21%
4000	0.007	0.004	0.002	12%	20%
4200	0.0073	0.003	0.0009	18.06%	26.88%
4900	0.0057	0.002	0.0012	18.13%	22.05%
5000	0.0064	0.0032	0.0016	16%	24%
5500	0.0051	0.0022	0.0018	15.95%	18.15%
6000	0.0046	0.0016	0.0016	18%	18%
6700	0.0041	0.0023	0.0005	12.06%	24.12%
7000	0.004	0.0017	0.0002	16.1%	26.6%
Success Ratio Average :				**15.8%**	**22.9%**

نُلاحظ من تحليل نتائج الاختبار الثاني أن الخوارزمية المُحسنة المقترحة MLFQ-EDF حسَّنت نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio بنسبة 22.9% عن خوارزمية MLFQ وبنسبة 15.8% عن خوارزمية MLFQ-RT ويوضح الشكل (3-6) نسب التحسن.



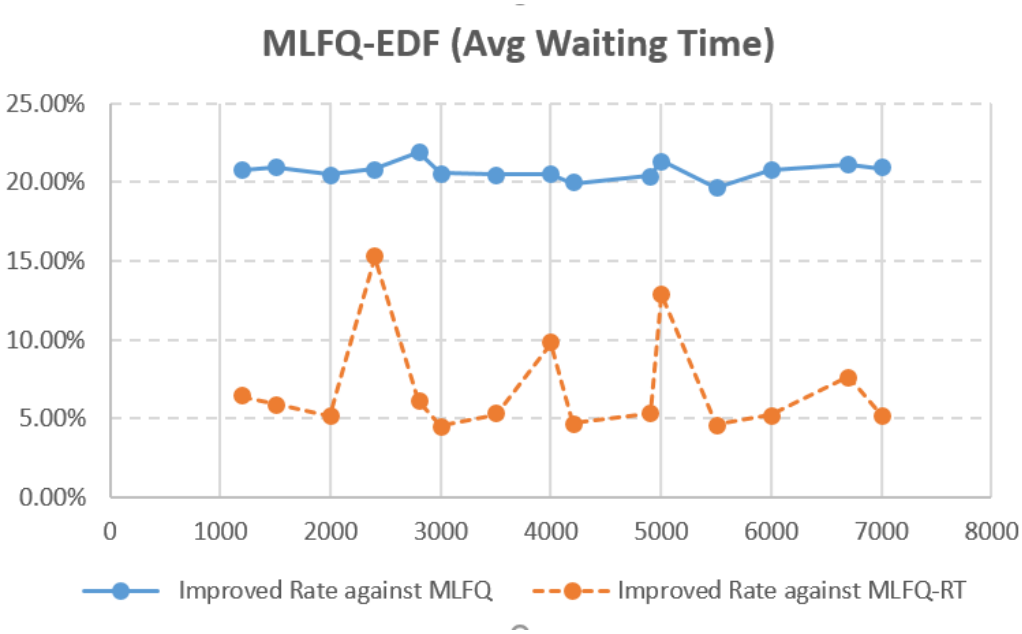
الشكل (3-6): تحسن نسبة نجاح إجرائيات الزمن الحقيقي (الاختبار الثاني) MLFQ-EDF.

ويوضح الجدول (3-21) مقارنة متوسط أزمنة انتظار الإجرائيات Avg Waiting Time في الخوارزمية المقترحة MLFQ-EDF مع كل من الخوارزميتين MLFQ و MLFQ-RT

الجدول (0-20): مقارنة نسب تحسن متوسط أزمنة انتظار الإجرائيات (الاختبار الثاني)

Co. of Processes	**Avg Waiting Time Improved Algorithm (MLFQ-EDF)** (ms)	**Avg Waiting Time (MLFQ-RT)** (ms)	**Avg Waiting Time (MLFQ)** (ms)	**Improved Rate (MLFQ-RT) %**	**Improved Rate (MLFQ) %**
1200	5581	5938	6740	6.39%	20.76%
1500	6839	7243	8270	5.90%	20.92%
2000	9138	9609	11012	5.15%	20.50%
2400	10880	12540	13147	15.25%	20.83%
2800	12845	13626	15658	6.08%	21.89%
3000	13765	14388	16601	4.52%	20.60%
3500	16220	17072	19541	5.25%	20.47%
4000	18359	20150	22131	9.75%	20.54%
4200	19468	20384	23348	4.70%	19.93%
4900	22641	23844	27258	5.31%	20.39%
5000	22942	25890	27848	12.84%	21.38%
5500	25851	27042	30931	4.60%	19.65%
6000	27456	28884	33160	5.20%	20.77%
6700	30769	33120	37279	7.64%	21.15%
7000	32265	33938	39005	5.18%	20.88%
Average :				**6.92%**	**20.7%**

نُلاحظ من تحليل نتائج الاختبار الثاني أن الخوارزمية المُحسنة المقترحة MLFQ-EDF حسنت معيار متوسط أزمنة انتظار الإجرائيات Avg Waiting Time بنسبة 20.7% عن خوارزمية MLFQ وبنسبة 6.92% عن خوارزمية MLFQ-RT المقترحة في الورقة البحثية [1] ويوضح الشكل (3-7) نسب التحسن.



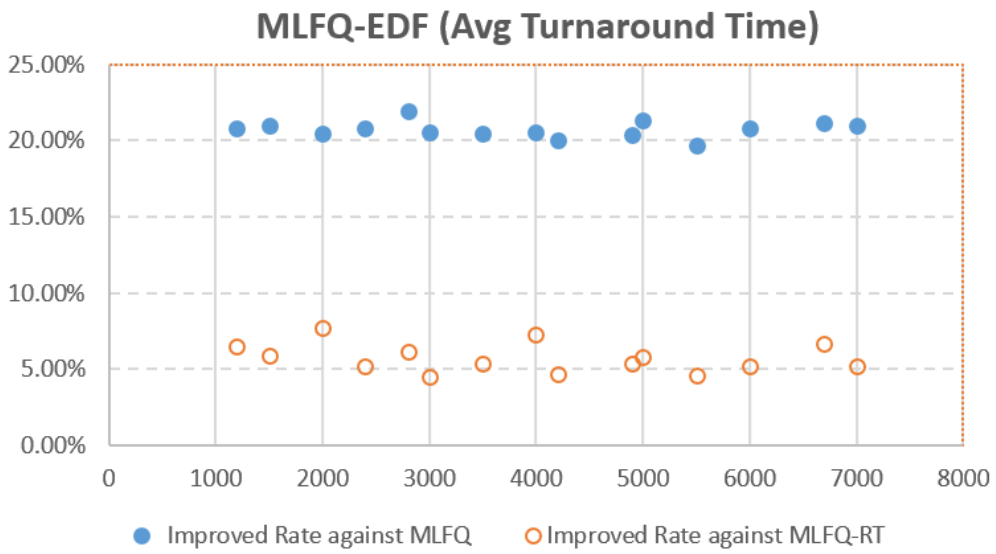
الشكل(3-7): 0تحسين معيار متوسط أزمنة انتظار الإجرائيات (الاختبار الثاني) MLFQ-EDF

ويوضح الجدول (3-22) مقارنة متوسط أزمنة التنفيذ الكلية للإجرائيات Avg Turnaround Time في الخوارزمية المقترحة MLFQ-EDF مع كل من الخوارزميتين MLFQ و MLFQ-RT .

الجدول (0-21): مقارنة نسب تحسن متوسط أزمنة التنفيذ الكلية للإجرائيات (الاختبار الثاني)

Co. of Processes	**Avg Turnaround Time Improved Algorithm (MLFQ-EDF)** (ms)	**Avg Turnaround Time (MLFQ-RT)** (ms)	**Avg Turnaround Time (MLFQ)** (ms)	**Improved Rate (MLFQ-RT) %**	**Improved Rate (MLFQ) %**
1200	5590	5947	6749	6.38%	20.73%
1500	6847	7251	8278	5.90%	20.89%
2000	9147	9855	11020	7.74%	20.47%
2400	10889	11442	13156	5.07%	20.81%
2800	12854	13635	15666	6.07%	21.87%
3000	13773	14396	16610	4.52%	20.59%
3500	16229	17081	19549	5.24%	20.45%
4000	18367	19690	22140	7.20%	20.54%
4200	19477	20393	23357	4.70%	19.92%
4900	22650	23852	27267	5.30%	20.38%
5000	22951	24273	27857	5.76%	21.37%
5500	25860	27051	30939	4.60%	19.64%
6000	27464	28892	33168	5.19%	20.76%
6700	30778	32820	37287	6.63%	21.14%
7000	32274	33946	39014	5.18%	20.88%
Average :				**5.7%**	**20.6%**

نُلاحظ من تحليل نتائج الاختبار الثاني أن الخوارزمية المُحسنة المقترحة MLFQ-EDF حسّنَت متوسط أزمنة التنفيذ الكلية للإجرائيات Avg Turnaround Time بنسبة 20.6% عن خوارزمية MLFQ وبنسبة 5.7% عن خوارزمية MLFQ-RT المقترحة في الورقة البحثية [1] ويوضح الشكل (3-8) نسب التحسن.



الشكل(3-8): تحسن متوسط أزمنة التنفيذ الكلية للإجرائيات (الاختبار الثاني) MLFQ-EDF

نسب التحسين للاختبار الثاني على المجموعة الثانية:

يوضح الجدول (3-23) نسب تحسن الخوارزمية المقترحة MLFQ-EDF عن كل من الخوارزميتين MLFQ وMLFQ-RT وفق معايير تقييم الأداء للاختبار الثاني.

الجدول (0-22): نسب التحسين في الاختبار الثاني

المعيار	معدل تَحسن خوارزمية MLFQ-EDF عن MLFQ-RT	معدل تَحسن خوارزمية MLFQ-EDF عن MLFQ
معدل نجاح إجرائيات الزمن الحقيقي Success Ratio	15.8 %	22.9 %
متوسط أزمنة انتظار الإجرائيات Avg Waiting Time	6.92 %	20.7 %
متوسط أزمنة تنفيذ الإجرائيات Avg Turnaround Time	5.7 %	20.6 %

مما سبق من خلال تحليل النتائج ونسب التحسن في الجدول (3-16) والجدول (3-23)، نلاحظ أن الخوارزمية المُقترحة MLFQ-EDF حققت تحسناً واضحاً في نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio ومتوسط أزمنة انتظار الإجرائيات Avg Waiting Time ومتوسط أزمنة التنفيذ الكلية للإجرائيات Avg Turnaround Time.

النتائج والآفاق المستقبلية

تم في هذا البحث اقتراح خوارزمية جدولة جديدة تدعى MLFQ-EDF كتحسين لخوارزمية جدولة التغذية العكسية متعددة المستويات MLFQ في أنظمة الزمن الحقيقي، من خلال استخدام خوارزمية جدولة في الزمن الحقيقي وهي خوارزمية الزمن النهائي الأقرب أولاً EDF ضمن خوارزمية MLFQ، حيث تم تنجيز خوارزمية MLFQ وخوارزمية MLFQ–RT المقترحة في [1] والخوارزمية المُقترحة في البحث MLFQ-EDF باستخدام لغة C++ على بيئة التطوير Visual Studio.

ثم اختُبرَت الخوارزميات على مجموعتان معياريتان من الإجرائيات المولدة عشوائياً بحيث تكون نسبة نجاح إجرائيات الزمن الحقيقي Success Ratio أكبر مايمكن، ومتوسط أزمنة انتظار الإجرائيات Avg waiting time ومتوسط أزمنة التنفيذ الكلية للإجرائيات Avg turnaround time أصغر مايمكن.

بعد تحليل نتائج الاختبارات المُنفذة على كل من خوارزميات الجدولة السابقة، أعطت الخوارزمية المُحسنة المقترحة MLFQ-EDF نتائج أفضل من كلتا الخوارزميتين، حيث حَسَّنت نسبة نجاح إجرائيات الزمن الحقيقي بنسبة 25% عن خوارزمية MLFQ، وبنسبة 17.3% عن خوارزمية MLFQ-RT.

إضافةً إلى تقليل متوسط أزمنة انتظار الإجرائيات بنسبة 23% عن خوارزمية MLFQ، وبنسبة 8.72% عن خوارزمية MLFQ-RT، وتقليل متوسط أزمنة التنفيذ للإجرائيات بنسبة 24% عن خوارزمية MLFQ، وبنسبة 8% عن خوارزمية MLFQ-RT.

نقترح كأعمال مستقبلية تحسين متوسط أزمنة الاستجابة للإجرائيات عن خوارزمية MLFQ، ودراسة إمكانية تنقيل إجرائيات الزمن الحقيقي RT ضمن أرتال خوارزمية MLFQ مع الأخذ بعين الاعتبار الموعد النهائي للإجرائيات Deadline، وإمكانية إدراج خوارزمية جدولة أخرى من خوارزميات الجدولة في الزمن الحقيقي ضمن خوارزمية جدولة رتل التغذية العكسية متعدد المستويات MLFQ وذلك لمعالجة إجرائيات الزمن الحقيقي.

المراجع

[1] Hoganson, K., & Brown, J. (2017, June). Real-time scheduling with MLFQ-RT multilevel feedback queue with starvation mitigation. In 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC) (pp. 155-160). IEEE.

[2] Hoganson, K. E. (2018, March). Multi-core real-time scheduling in multilevel feedback queue with starvation mitigation (MLFQ-RT). In Proceedings of the ACMSE 2018 Conference (pp. 1-6).

[3] AbdAlGhani, N., Mohamad, M., Atef, M., & Amgad, K. (2020). Developed Dynamic Multilevel Feedback Queue Scheduling Algorithm with Aging Criteria for Starvation Mitigation. 10.13140/RG.2.2.16046.54082.

[4] Abirami, B., & Vasudevan, V. (2023). Modified Multilevel Feedback Queue Scheduling Algorithm with Starvation Mitigation for Reconfigurable Computing systems.

[5] Sattar, I., Shahid, M., & Yasir, N. (2014). Multi-level queue with priority and time sharing for real time scheduling. International journal of multidisciplinary sciences and engineering, 5(8), 16-17.

[6] Hoganson, K., & Brown, J. (2017, April). Intelligent mitigation in multilevel feedback queues. In Proceedings of the SouthEast Conference (pp. 158-163).

[7] Thombare, M., Sukhwani, R., Shah, P., Chaudhari, S., & Raundale, P. (2016, March). Efficient implementation of multilevel feedback queue scheduling. In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 1950-1954). IEEE.

[8] Harki, N. (2024). Multi-Level Feedback Queue Scheduling Technique. Cihan University-Erbil Scientific Journal, 8(2), 36-42. https://doi.org/10.24086/cuesj.v8n2y2024. pp36-42

[9] Gupta, A. K., Yadav, N. S., & Goyal, D. (2021). Design and performance evaluation of smart job first multilevel feedback queue (SJFMLFQ) scheduling algorithm with dynamic smart time quantum. In Research Anthology on Advancements in Quantum Technology (pp. 111-126). IGI Global.

[10] Sun, B., Kloda, T., Garcia, S. A., Gracioli, G., & Caccamo, M. (2024). Minimizing cache usage with fixed-priority and earliest deadline first scheduling. Real-Time Systems, 1-40.

[11] Rao, G. S. N., Kumar, J. R., & Rao, T. S. (2018, December). An Intelligent MLFQ Scheduler Prediction Using Neural Approach. In 2018 Tenth International Conference on Advanced Computing (ICoAC) (pp. 94-98). IEEE.

[12] Salomo, R., Ranggadara, I., Kurnianda, N. R., & Suhendra. (2019). Multilevel Algorithm Feedback Queue for Restaurant Order System. International Journal of Recent Technology and Engineering. International Journal of Recent Technology and Engineering (IJRTE) (pp. 5000-5005). ISSN: 2277-3878, Volume-8 Issue-3.

[13] Chetto, M., & El Osta, R. (2023). Earliest Deadline First Scheduling for Real-Time Computing in Sustainable Sensors. Sustainability, 15(5), 3972

[14] Mulder, D., Ssempala, J., Walton, T., Parker, B., Brough, S., Bush, S., Boroojerdi, S., & Tang, J. (2020). Impact of Quantum Values on Multilevel Feedback Queue for CPU Scheduling. 2020 Intermountain Engineering, Technology and Computing (IETC), (pp. 1-4). IEEE. https://doi.org/10.1109/IETC47856.2020.9249188

[15] Kashif, M., Nawaz, R., Rulaningtyas, R., Iqbal, S., & Zulfiqar, A. (2023). Multilevel feedback queue: Efficient scheduling and implementation by using dynamic quantum. The 8th International Conference and Workshop on Basic and Applied Science (ICOWOBAS). AIP Conf. Proc. 2554, 040004. https://doi.org/10.1063/5.0106240

[16] Pathan, R.M. (2016). Design of an efficient ready queue for earliest-deadline-first (EDF) scheduler. 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), 293-296.

[17] Zhao, X., Wei, Y.D., & Li, W. (2017). The Improved Earliest Deadline First with Virtual Deadlines Mixed-Criticality Scheduling Algorithm. 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), 444-448.

[18] Abeni, L., & Cucinotta, T. (2020). EDF scheduling of real-time tasks on multiple cores. ACM SIGAPP Applied Computing Review, 20, 5 - 18. Volume 20, Issue 2. Pages 5-18. https://doi.org/10.1145/3412816.3412817

[19] Silberschatz, A., Galvin, P. B., Gagne, G. (2018). Operating System Concepts (10th ed.). United Kingdom: Wiley.

[20] Stallings, W. (2016). Computer organization and architecture: Designing for performance (10th ed.). United Kingdom: Pearson Education Limited.

[21] Tanenbaum, A. S., & Bos, H. (2014). Modern operating systems (4th global ed.). United Kingdom: Pearson Education Limited.

[22] Stankovic, J. A., Spuri, M., Ramamritham, K., & Buttazzo, G. (1998). Deadline scheduling for real-time systems: EDF and related algorithms (Vol. 460). Springer Science & Business Media.

[23] Wu, X., & Loiseau, P. (2024, April). Algorithms for scheduling deadline-sensitive malleable tasks. In Operations Research Forum (Vol. 5, No. 2, p. 30). Cham: Springer International Publishing.

[24] Satyanarayana, S.K., Kumar, P.S., & Sridevi, G. (2018). Improved Process Scheduling in Real-Time Operating Systems Using Support Vector Machines. Proceedings of 2nd International Conference on Micro-Electronics, Electromagnetics and Telecommunications. Lecture Notes in Electrical Engineering, vol 434. Springer, Singapore. https://doi.org/10.1007/978-981-10-4280-5_63

[25] Cottet, F. (2002). Scheduling in Real-Time Systems. United Kingdom: Wiley.

Abstract

Task scheduling is crucial in computer systems, and selecting an appropriate process scheduling algorithm to ensure minimal waiting time and execution time is highly significant. In Real-Time Systems, the focus is on guaranteeing that tasks meet critical deadlines by scheduling all tasks with a maximum success ratio for real-time processes compared to non-real-time processes.

This research proposes the MLFQ-EDF scheduling algorithm as an improvement to one of the CPU scheduling algorithms, the Multilevel Feedback Queue (MLFQ) algorithm, to improve the performance of this algorithm by increasing the Success Ratio and reducing the average waiting time and the average total execution time.

The proposed algorithm combines two scheduling algorithms, the Earliest Deadline First (EDF) algorithm and the MLFQ algorithm, where the EDF algorithm is applied in the highest-level ready queue of the MLFQ algorithm in order to obtain the best Success Ratio for real-time processes and the lowest average waiting time and total execution time by comparing the implementation and experimentation of the proposed MLFQ-EDF algorithm with the MLFQ algorithm and its previously improved version, MLFQ-RT.

Each algorithm was implemented and simulated using C++ and Visual Studio environment.

The results, after implementing and testing the proposed MLFQ-EDF algorithm, showed a 25% improvement in the success ratio of real-time processes, a 23% improvement in average waiting time, and a 24% improvement in average turnaround time compared to the MLFQ algorithm. Additionally, there was a 17.3% improvement in the success ratio of real-time processes, an 8.72% improvement in average waiting time, and an 8% improvement in average turnaround time compared to the previously improved MLFQ-RT algorithm.

Syrian Arab Republic
Damascus University
Faculty of Mechanical & Electrical Engineering
Department of Computer & Automation Enginering

Improving the Multilevel Feedback Queue (MLFQ) Algorithm in Real Time Systems

A thesis prepared for the degree of Master of Science in Computer Engineering and Networks

By:

Eng. Bassel Ali Assad

Supervison By:

Dr. Eng. Rafaah Khazem

Professor in Department of Computer and Automation Engineering

2024−2023