مجلة جامعة طرطوس للبحوث والدراسات العلمية_ سلسلة العلوم الهندسية المجلد (٤) العدد (٤) 2020
Tartous University Journal for Research and Scientific Studies- Engineering Sciences Series Vol. (4) No. (4) 2020

المهام الاستباقية والمجدولة في نظم الاتصالات

غسان محمد محمد*

(تاريخ الإيداع ١١ / ٣ / ٢٠٢٠ . قُبل للنشر ١٧ / ٦ / ٢٠٢٠)

الملخص

يتضمن البحث دراسة وجود المهام الاستباقية "والتي تكون ذات اولوية عالية" والمهام المجدولة "والتي تبدأ في وقت محدد كنتيجة لحدث معين" وذلك من خلال محاكاة ارتال تعتمد الاولوية في عملها باستخدام بيئة الماتلاب و تم التوصل للعديد من النتائج التي تمكننا من فهم كيفية التعامل مع المهام المجدولة والاستباقية في اي نظام اتصالات .

كلمات مفتاحية : مهام ، مهام استباقية ، مهام مجدولة ، نظام اتصالات ، اولوية ، رتل ، رتل اولوية

*أستاذ مساعد – قسم هندسة تكنلوجيا الاتصالات – كلية هندسة تكنلوجيا الاتصالات و المعلومات – جامعة طرطوس – طرطوس – سورية.

مجلة جامعة طرطوس للبحوث والدراسات العلمية_ سلسلة العلوم الهندسية المجلد (٤) العدد (٤) 2020
Tartous University Journal for Research and Scientific Studies- Engineering Sciences Series Vol. (4) No. (4) 2020

Proactive and Scheduled Tasks in Communication Systems

Ghassan Mohammed Mohammed*

(Received 11 / 3 / 2020 . Accepted 17 / 6 / 2020)

ABSTRACT

The research involves examining the existence of proactive tasks that are "high priority", and scheduled tasks " which start at a specific time as a result of a particular event," by simulating priority queues in their work using the matlab environment and many results have been achieved that enable us to understand how to deal with scheduled and proactive tasks in any communication system.

Key Words : Tasks, proactive tasks, scheduled tasks, communication system, priority, queue

*Associate Professor, Department of Communications Engineering, Faculty of ICT Engineering, Tartous, University, Tartous, Syria.

مقدمة :

يتسم القرن الواحد والعشرون بتسارع كبير في وتيرة التطور التكنولوجي، حيث يشهد العالم بأكمله تفجراً معرفياً لافتاً وتطور عاصف في مجالات العلوم التكنولوجية والاتصالات، في ضوء هذا كله تعد المهام الاستباقية و المهام المجدولة ذات فائدة عملية كبيرة وتحل الكثير من المشكلات في اي نظام بشكل سريع واني وتنفيذ عمليات بشكل تلقائي كنتيجة لاي مشكلة تحدث في اي نظام اتصالات من خلال استغلال الموارد المتوفرة بافضل شكل للحصول على افضل اداء ممكن سواء كانت المنظومة تعاني مشاكل او لا .

هدف البحث وأهميته :

تنحصر الغاية الأساسية من هذا البحث في دراسة ومحاكاة المهام الاستباقية والتي تهدف الى القيام بمهمة طارئة او لها اولوية عالية جدا لتقوم بعملية مقاطعة لمهام قيد التنفيذ ذات اولوية طبيعية و هذا يعد مفيد في حل المشاكل في اي نظام قبل تأزمها وخروجها عن السيطرة من خلال تنفيذ مهام استباقية طارئة (او يجب تنفيذها في الوقت الفعلي على وجه السرعة ولا تحتمل اي تأخير) وكذلك دراسة ومحاكاة المهام المجدولة والتي يلخص الهدف منها في انها تنفذ عند تنفيذ مهام اخرى سابقة حيث انه من وظائف اي نظام اتصالات جدولة استخدام المصادر و استغلالها بشكل أمثل بحيث أن نظام الاتصالات يحدد المهام المطلوبة و المصادر المتوفرة من معالج و ذاكرة و أجهزة في كل لحظة زمنية و يوزع عليها المهام المطلوبة بطريقة تزيد من سرعة انجاز العمل .

طريقة البحث :

اعتمد في هذا البحث منهجية المحاكاة الحاسوبية وتم اعتماد التسلسل التالي في البحث :

- لمحة عامة عن المهام الاستباقية والمجدولة .
- لمحة وتعريف بنظام الارتال ذو الاولوية .
- الحالات الاساسية للدخول الى الرتل و المعالجة .
- القيام بعملية المحاكاة من خلال دارة تركز على المهام الاستباقية وتحوي جدولة مهام مع شرح مكونات الدارة .
- القيام بعملية المحاكاة من خلال دارة تركز على المهام المجدولة مع شرح مكونات الدارة .
- اظهار نتائج المحاكاة وتفسير النتائج .

١- نظرية الارتال :

هي دراسة رياضية لصفوف (ارتال) الانتظار والتي تمكن من التحليل الرياضي للعديد من العمليات كتلك الواردة الى الرتل والانتظار في الرتل (عملية تخزين أساساً) وخدمة من في مقدمة الرتل من قِبل المعالج او عدة معالجات حيث تسمح هذه النظرية باشتقاق وحساب العديد من مقاييس الأداء، بما في ذلك معدل وقت الانتظار في الرتل أو النظام

والعدد المتوقع ممن ينتظرون أو يتلقون الخدمة واحتمال مواجهة النظام لبعض الحالات كأن يكون فارغاً أو ممتلئاً أو وجود معالج شاغر أو الانتظار لبعض الوقت حتى يمكنه تلقي العملية .

لا يمكن للرتل سوى تغيير ترتيب المهام المنتظرة بداخله بحيث يكون العدد الاجمالي للمهام المنتظرة بداخله لا تتأثر فبالتالي مجموع متوسط أوقات الانتظار للمهام المختلفة المنتظرة بداخله دائما ثابت وبالتالي أي محاولة لتحسين الخدمة لبعض المهام ذات اولوية اعلى يجب أن يأتي على حساب المهام ذات الاولوية الاقل الاخرى [1].

٢- الحالات الاساسية للدخول الى الرتل و المعالجة :

٢-١- The M/G/1 conservation law :

اثبت كلينروك انه يجب على اي نظام M/G/1 غير استباقي ان [2] :

$$\sum_{k=1}^{K}\rho_k m_k = \begin{cases} W_0/(1-\rho) - W_0 = \rho W_0/(1-\rho), & \rho < 1; \\ \infty, & \rho \geq 1; \end{cases} \qquad \text{where } \rho_k = \lambda_k/\mu_k \text{ and } \rho = \sum_{k=1}^{K}\rho_k \qquad (١)$$

حيث W_0 يمثل زمن الحياة المتبقية للمهمة ، ويعطى وفق القانون التالي : $$W_0 = \sum_{k=1}^{K}\frac{\lambda_k \overline{x_k^2}}{2} \qquad (٢)$$

٢-٢- The G/G/1 conservation law :

في حالة وجود نظام طوابير محدد G / G / 1 ذو اولوية فاننا سنربطه بقانون خطي وذلك بغض النظر عن انضباط الدور في الطابورالدور [3]:

$$\sum_{k=1}^{K}\rho_k m_k = \bar{U} - W_0 \qquad (3)$$

حيث : $\tilde{u}$ المتوسط المحدد بالمهام غير المنجزة في النظام

في M/G/1 قيمة $\tilde{u}$ تعطى بالعلاقة $W_0/(1-\rho)$ التالية : (٤)

٢-٣- The G1/G/1 conservation law :

في حال وجود نظام طوابير multi-class فانه يوجد ثابت V يحدد فقط التوزع الرياضي لزمن الوصول والخدمة مثال على ذلك [4]:

$$\sum_{k=1}^{K}\rho_k m_k = V + \sum_{k=1}^{K}\rho_k\left(\frac{1}{\mu_k} - \gamma_k\right) \qquad (٥)$$

حيث $\gamma_k = \mu_k\overline{x_k^2}/2$ ان (٦) حيث k لها قيمة ١,٢,.....,k وهي تعبر عن متوسط الزمن المتبقي حتى يتم خدمة المهمة ذات الاولوية k .

٣- الارتال ذات الاولوية :

تعتبر بنية الرتل ذات الاولوية من بنى المعطيات التي يمكن استخدامها لتخزين المهام التي لها اولويات مختلفة او متشابهة فهذا النوع من الارتال يحتوي على مقدمة و مؤخرة وتكون المهام بداخله مرتبة بحسب اولويتها بحيث المهام ذات الاولويات الاعلى تعالج قبل المهام ذات الاولويات المنخفضة .

تنقسم الارتال ذات الاولويات الى نوعين : تصاعدية الترتيب (ascending-priority queue) وتنازلية الترتيب (descending-priority queue) [2] .

٣- المهام الاستباقية في النظام :

في عمليات المعالجة و الحوسبة ، يكون الاستباق هو بمثابة تعطيل مؤقت للمهمة التي يقوم بها المعالج وبغرض استئناف المهمة في وقت لاحق . تُعرف هذه التغييرات في المهمة التي تم تنفيذها كمبدلات السياق (context changes) . وعادة ما يتم ذلك من خلال مهمة متميزة أو جزء من النظام يعرف باسم المجدول الوقائي (Preventive scheduler) ، والذي لديه القدرة على استباق أو مقاطعة (Pre-empt or interrupt) ، واستئناف مهام أخرى في النظام في وقت لاحق .

يستخدم مصطلح تعدد المهام الوقائي لتمييز نظام متعدد المهام ، والذي يسمح باستباق المهام في نظام متعدد المهام حيث يجب برمجة العمليات أو المهام بشكل صريح حتى لا تحتاج إلى موارد النظام.

بعبارة بسيطة: يشمل تعدد المهام الوقائية (Preventive tasks) استخدام آلية المقاطعة التي تعلق عملية التنفيذ حاليًا وتستدعي جدولة لتحديد العملية التي يجب تنفيذها بعد ذلك. لذلك ستحصل كل العمليات على بعض الوقت من وحدة المعالجة المركزية في أي وقت معين.

في تعدد المهام الوقائي ، يمكن لنواة نظام التشغيل أيضًا بدء تبديل السياق لتلبية قيود أولوية السياسة المجدولة ، وبالتالي استباق المهمة النشطة. بشكل عام ، يعني الاستباق "الاستيلاء المسبق على". عندما تنتهي المهمة ذات الأولوية العالية بعد ان تكون قد استبقت المهمة قيد التشغيل حاليًا ، فإنها تُعرف باسم الجدولة الوقائية (الاستباق) .

أحيانًا يتم استخدام المصطلح "تعدد المهام الوقائي" عن طريق الخطأ عندما يكون المعنى المقصود أكثر تحديدًا ، فيشير بدلاً من ذلك إلى فئة سياسات الجدولة المعروفة باسم الجدولة الزمنية المشتركة أو تقاسم الوقت.

يسمح تعدد المهام الوقائي للنظام بأن يضمن بشكل اكبر الموثوقية (Reliability) من اجل كل عملية في وقت تشغيل النظام . كما يسمح النظام بالتعامل بسرعة مع الأحداث الخارجية المهمة مثل البيانات الواردة ، والتي قد تتطلب الانتباه الفوري من عملية أو أخرى.

في أي وقت محدد ، يمكن تجميع العمليات إلى فئتين: تلك التي تنتظر الإدخال أو الإخراج (تسمى " I / O bound") ، و العديد من العمليات التي تستخدم وحدة المعالجة ("CPU bound"). في الأنظمة التقليدية القديمة ، غالبًا ما كانت تقوم "بلخمول" أثناء انتظار الإدخال المطلوب وخلال هذه الفترة ، لم تكن العملية تؤدي عملاً مفيدًا ، ولكنها لا تزال تحافظ على التحكم الكامل في وحدة المعالجة. مع ظهور المقاطعات وعمليات تعدد المهام الوقائية ، يمكن "منع" عمليات الإدخال / الإخراج المرتبطة ، أو تعليقها ، في انتظار وصول البيانات الضرورية ، مما يسمح للعمليات الأخرى باستخدام وحدة المعالجة. وبما أن وصول البيانات المطلوبة من شأنه أن يولد مقاطعة ، فيمكن ضمان العمليات المحظورة في الوقت المناسب للعودة إلى التنفيذ.

لا يتطلب تعدد المهام تنفيذ متوازي لمهام متعددة في نفس الوقت بالضبط بدلا من ذلك فإنه يسمح لأكثر من مهمة واحدة للتقدم للمعالجة خلال فترة زمنية معينة وحتى في الانظمة ذات المعالجات المتعددة ، فإن تعدد المهام يسمح بتشغيل العديد من المهام أكبر من عدد وحدات المعالجة (CPUs).

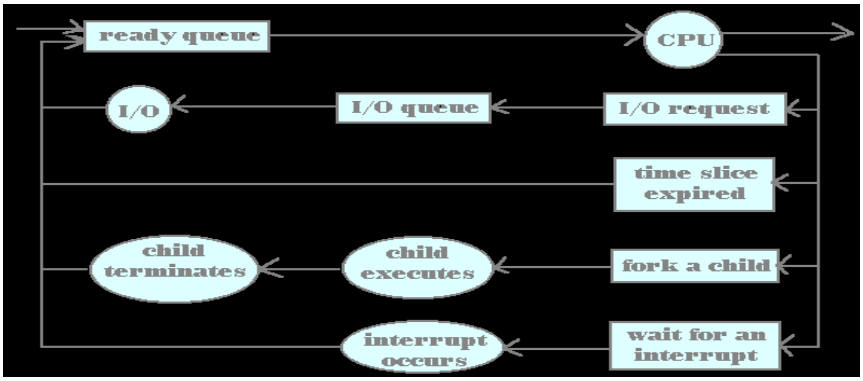
يوضح هذه المحاكاة التي ستلي هذا الشرح النظري جدولة المهام وتنفيذها في نظام تشغيل متعدد المهام. فهو لا يوضح فقط جانب استباقية المهام ولكنه يوضح أيضًا كيفية محاكاة تحديد أولويات الأجزاء الزمنية الحرجة وغير الحساسة للوقت في العملية باستخدام المهام التابعة [3].

تم محاكاة نظام يحتوي ارتال تعتمد الاولوية في عملها حيث فرضنا ان لدينا نظام باربع انواع من المهام ولها اولويات كما تم تنفيذه .

٥- المهام المجدولة في النظام :

من وظائف نظام التشغيل في نظم الاتصالات جدولة استخدام المصادر و استغلالها بشكل أمثل، حيث أن نظام التشغيل يحدد المهام المطلوبة و المصادر المتوفرة من معالج و ذاكرة و أجهزة في كل لحظة زمنية و يوزع عليها المهام المطلوبة بطريقة تزيد من سرعة انجاز العمل.

عندما يدخل البرنامج النظام فانه يوضع فيjob queue عن طريق برنامج queue routine وعندما تتوفر المساحة اللازمة في الذاكرة فان المجدول يختار إحدى المهام و يعمل لها تحميل في الذاكرة الرئيسية و توضع في ready queue و هناك ثلاثة طرق أساسية للجدولة: الجدولة طويلة الأجل والجدولة متوسطة الأجل والجدولة قصيرة الأجل .[4]



الشكل (١) مخطط صندوقي لالية عمل المهام المجدولة في النظام

٥-١- الجدولة طويلة الأجل long-term scheduling:

هي التي تحدد المهام التي سوف تدخل النظام لتنفيذها و أي المهام يجب أن تخرج من النظام حيث تقوم بنقل العمليات (process) من الـ disk إلى الذاكرة (memory) حسب توفر المكان في الذاكرة وتتم بين مرحلتي new processو ready process و تتحكم فيما يعرف بدرجة تعدد البرمجة(multiprogramming) وهي عدد العمليات في الذاكرة. و الجدولة طويلة الأجل مهمة في نظم تشغيل الوقت الحقيقي وهي النظم التي تتحكم بحواسيب تتفاعل مع البيئة لتنفيذ العمل بحيث تكون عملية الادخال والمعالجة في نفس اللحظة مثل أجهزة تخطيط القلب.

٥-٢- الجدولة متوسطة الأجل medium-term scheduling:

time-sharing systems و هذه الجدولة موجودة في بعض أنظمة التشغيل مثل: أنظمة اقتسام الزمن و تقوم هذه الجدولة بازالة المهام من الذاكرة ، لعدة أسباب منها أن تكون المهمة لها أولوية منخفضة، وهذه العمليةتسمى بالمقايضةswapping وهي التي تحدد أي المهام يجب أن تؤجل و أيها يجب أن تستأنف.

٥-٣- الجدولة قصيرة الأجل short-term scheduling :

هي التي تحدد أي المهام التي يمكن أن تستحوذ على موارد المعالج و الى أي مدة حيث تقوم بنقل العمليات الجاهزة للتنفيذ من الذاكرة إلى المعالج (cpu) ليتم تنفيذها وتتم بين مرحلتي ready process و running process ويجب أن تكون هذه العملية سريعة.

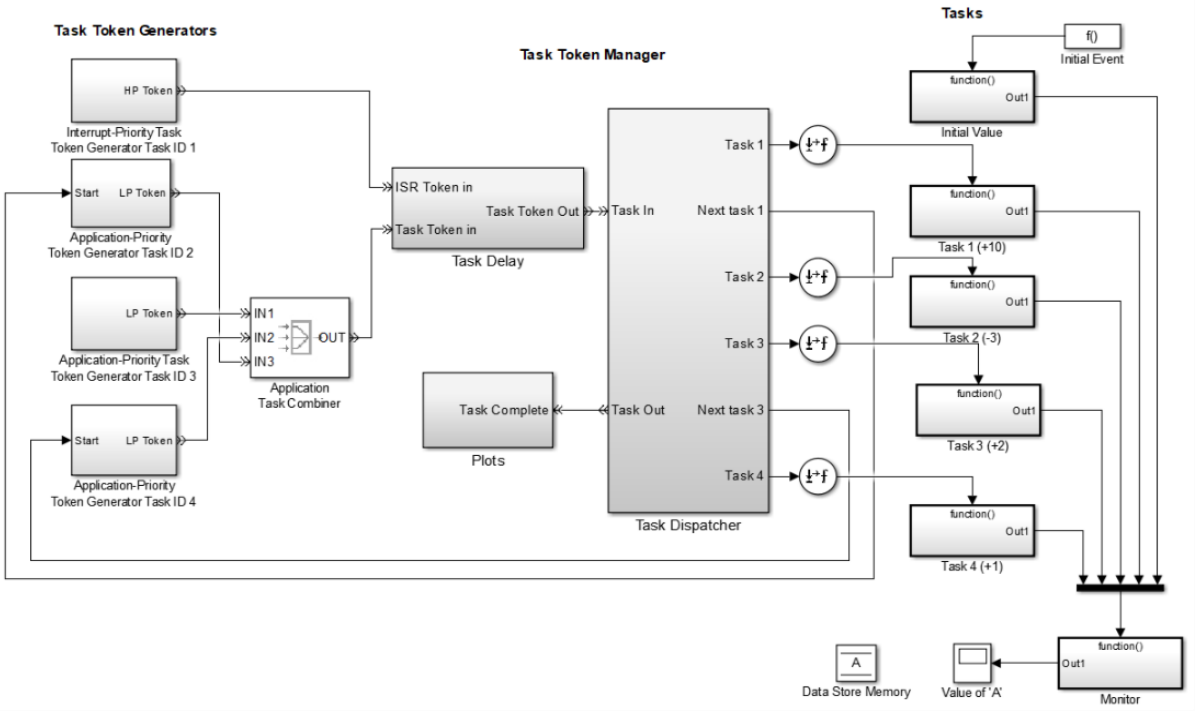
في بعض الأنظمة قد لاتكون موجودة الجدولة طويلة الأجل ففي ميكروسوفت ويونيكس ليس لديها الجدولة طويلة الأجل (long-term scheduling) حيث تقوم بوضع العمليات الجديدة فوراً في الذاكرة مما يعني جدولة قصيرة الأجل.

ان تبديل السياق context switch ميزة من مميزات نظام التشغيل متعدد المهام، وهي تجعل المعالج المركزي ينتقل من مهمة الى أخرى بدلا من تخصيص حيز زمني من أجل كل مهمة على التعاقب، وقبل أن تتم هذه العملية يجب تخزين حالة المهمة قبل الانتقال الى مهمة أخرى، مثلاً تخزين الحالة الحالية للمهمة(٠) ومن ثم تحميل الحالة التخزينية السابقة للمهمة(١).

وهناك ثلاث حالات يجب أن تحدث فيها عملية تبديل السياق [1]:

١. تعدد المهام. multitasking
٢. معالجة الانقطاع. interrupt handling
٣. التبديل بين نمط المستخدم و نمط النواة. user and kernel mode switching

٦- مثال عملي يركز على العمليات الاستباقية من خلال المحاكاة و فهم طبيعة العمل :



الشكل (٢) النموذج المعتمد لمحاكاة المهام الاستباقية في النظام

٦-١- قسم Task Token Generators :

في كل بلوك من هذا القسم يتم توليد ارقاما عشوائية بطريقة مستندة على الحدث اي يتم توليد كتلة ارقام عشوائية مختلفة من اجل كل مهمة تصل الى المعالج حيث ان كتلة الارقام العشوائية التي يتم توليدها من اجل كل مهمة تسحب من تابع توزع معين [5] وهنا تم الاعتماد على تابع توزع **Uniform والذي يعطى بلعلاقة التالية** :

$$f(x) = \begin{cases} \frac{1}{U-L} & \text{for } L \le x \le U \\ 0 & \text{otherwise} \end{cases} \quad (٧)$$

حيث L هو البارمتر الاقل قيمة و u هو البارمتر الاعلى قيمة .

يتم توليد المهمات بفاصل زمني محدد باستخدام مولد الكينونات (Generator of entities) المعتمد على الزمن ثم يتم تعيين سمات محددة لكل هذه المهمات منها الاولوية و الزمن اللازم لتنفيذ المهمة و معرف المهمة ويتم ربط كل مهمة برقم عشوائي يسحب من تابع توزع احتمالي كما اسلفنا مع ملاحظة انه يشير وجود منفذ دخل الى مولد الكينونات إلى أن هذا النوع من المهام يعتمد على إكمال نوع آخر من المهام.

المهام التي يتم توليدها في هذا القسم سيتم تصنيفها حسب المعرف الخاص بها الى :

١ – مهمة أولوية المقاطعة ذات الاولية العليا بلتاكيد

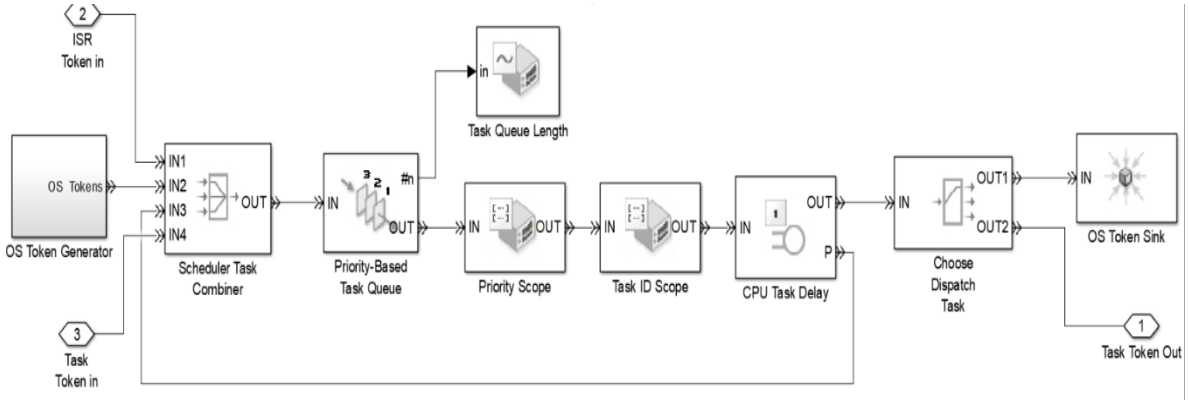
٢ – مهمة أولوية تطبيق معين التي تعتمد على إتمام المهمة ١

٣ – مهمة أولوية تطبيق معين

٤ – مهمة أولوية تطبيق معين التي تعتمد على إتمام المهمة ٣

٦-٢- قسم Task Delay :

يظهر الرسم التالي مكونات هذا القسم



الشكل (٣) قسم Task Delay في نموذج المحاكاة

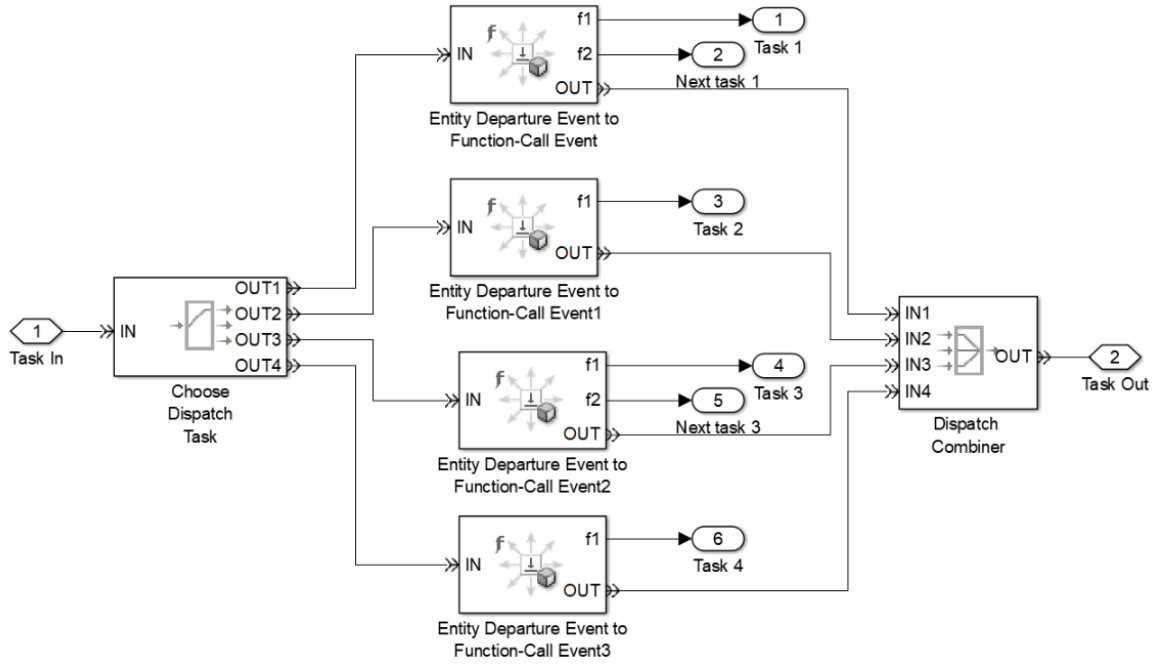
تصل المهام الى الرتل ذو الاولويات والذي يعمل بالأولوية التصاعدية ثم التقدم إلى المعالج المسمى CPU Task Delay . يمكن لمهمة ذات أولوية أعلى استباق (مقاطعة) مهمة ذات أولوية أقل. عند الاستباق ، تسجل المهمة ذات الأولوية المنخفضة وقت التنفيذ المتبقي وترجع إلى قائمة الانتظار(الرتل) ولا بد من ملاحظة ان المهام التي تصل الى هذا القسم يتم جدولتها من خلال مجدول لتصل الى الرتل من خلال مسار واحد .

في اللحظة $T = 0$ ، يقوم نظام OS Token Generator الفرعي بتوليد مهمة الخمول حيث ان مهمة الخمول لها مدة غير محدودة وأولوية منخفضة وبمعرف مهمة له قيمة الصفر و عندما يتم اكتشاف ان المهمة الواصلة هي مهمة خمول ، فإن ذلك يشير إلى عدم وجود مهمة قيد التنفيذ حاليًا.

بعد خروج المهام من المعالج فان المهمة تكون قد تعرضت لتاخير اضافي ويتم بعد ذلك ارسالها الى المصب اذا كانت مهمة خاملة والا فانها ستخضع لمراحل اخرى في النظام كما سنرى .

٦-٣- قسم Task Dispatcher :

يظهر الرسم التالي مكونات هذا القسم



الشكل (٤) قسم Task Dispatcher في نموذج المحاكاة

باستخدام المعرف الخاص بلمهمة الواصلة الى هذا القسم ، يقوم مرسل المهام بتوجيه كل مهمة إلى المحول الذي يقوم بترجمة حدث المغادرة للكيان إلى وظيفة أو وظيفتين حيث في مثالنا هذا تقوم هذه الكتلة بإنشاء استدعاء دالة أو اثنين .

نلاحظ ان المهمة ذات المعرف رقم واحد والتي تعتمد في تنفيذها على المهمة رقم اثنان تحتاج الى تنفيذ وظيفتين وهما وظيفة المهمة الاولى و الاخرى الاتصال مع Task Token Generator لتوليد مهمة بمعرف ذو رقم اثنان والذي تعتمد عليه المهمة ذات الرقم واحد .

٦-٤- قسم Tasks :

يتم تنفيذ النظام الفرعي المناسب الذي يمثل الإجراء (الوظيفة) الذي تم تنفيذه في تلك المهمة حيث في هذا النموذج تكون الاجراءات عبارة عن عمليات حسابية على الإشارة A وهنا يمكن تطوير النموذج بتنفيذ اي اجرائات اخرى والتي يمكن أن تكون أكثر تعقيدًا وبالتالي يفتح هذا البحث المجال لاي تطويرات مستقبلية عليه [6] .

تتم العملية الحسابية على الاشارة A تتم وفق العمليات الواردة للمعالج ومن ثم تفسيرها حيث تم برمجة الذاكرة بحيث تخزن قيمة A مبدئيا بلعدد ٥ وبرمجة عمليات حسابية تؤثر على قيمة الاشارة بالاضافة او النقصان حسب المهمة

المراد تفسيرها حيث : المهمة ذات المعرف رقم واحد تعني اضافة الرقم ١٠ الى قيمة الاشارة

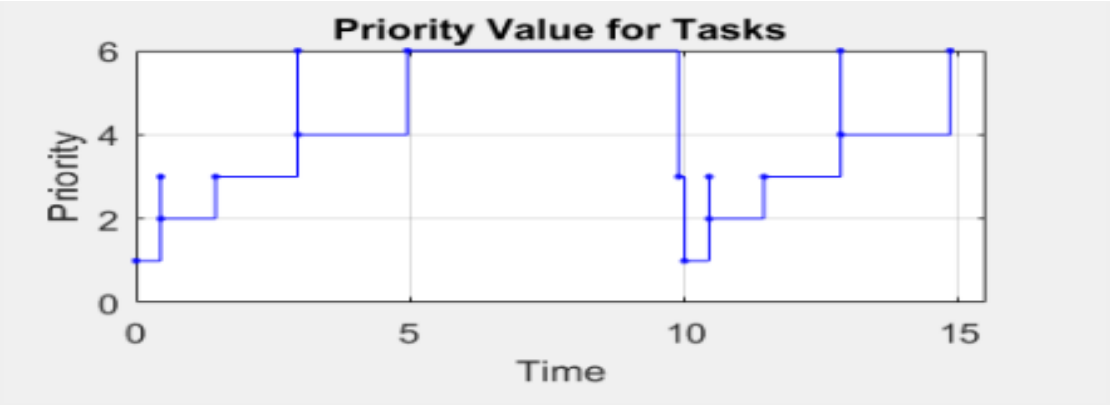
المهمة ذات المعرف رقم اثنان تعني طرح الرقم ٣ من قيمة الاشارة

المهمة ذات المعرف رقم ثلاثة تعني اضافة الرقم ٢ الى قيمة الاشارة

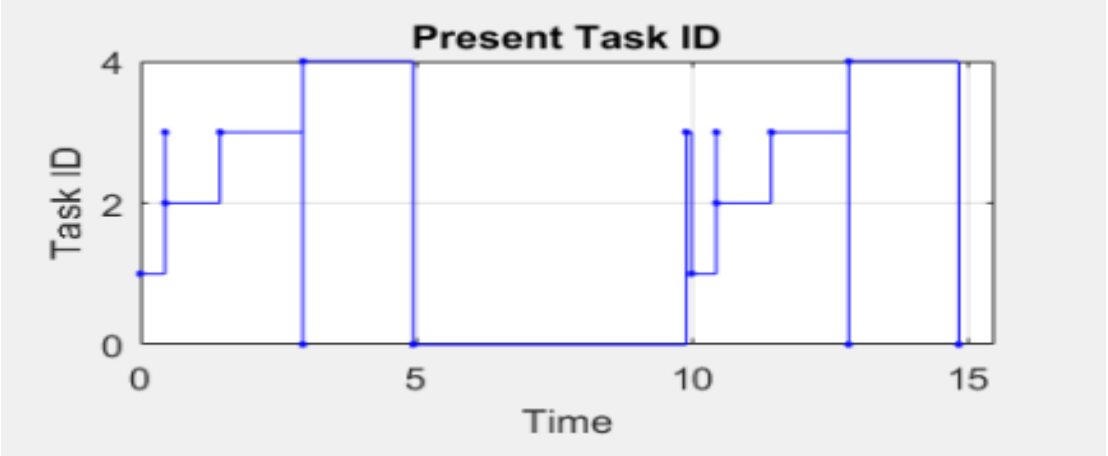
المهمة ذات المعرف رقم اربعة تعني اضافة الرقم ١ الى قيمة الاشارة

لا بد من التنويه الى ان الاجراء الذي يتم توليده عند ورود مهمة الى المحول في قسم Task Dispatcher هي اجراء معتمد على الحدث لذلك نضع على كل منفذ function من اجل كل محول محول من اجراء معتمد على الحدث الى اجراء معتمد على الزمن لكي نستطيع ان نرسم الخطوط البيانية ونفسر النتائج والقيام بالحسابات المطلوبة .

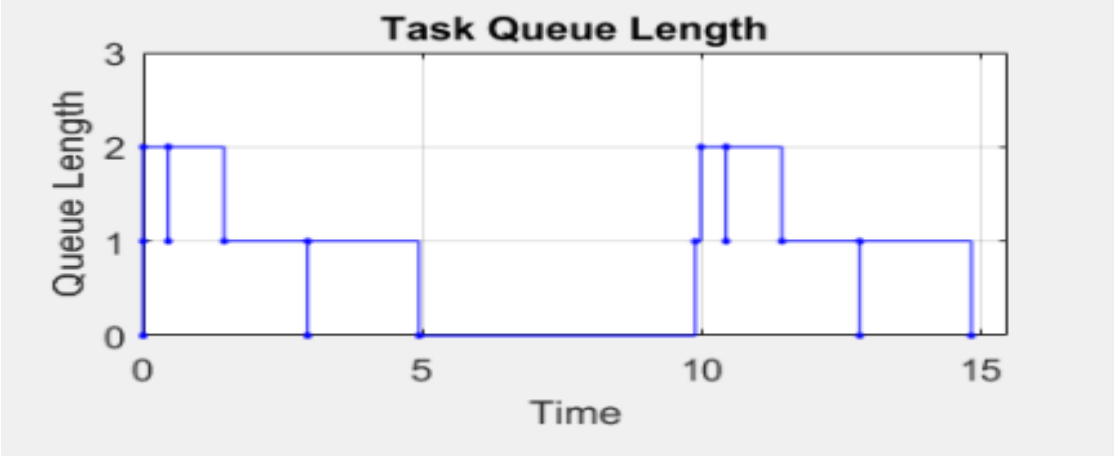
٧- المناقشة والاستنتاجات:



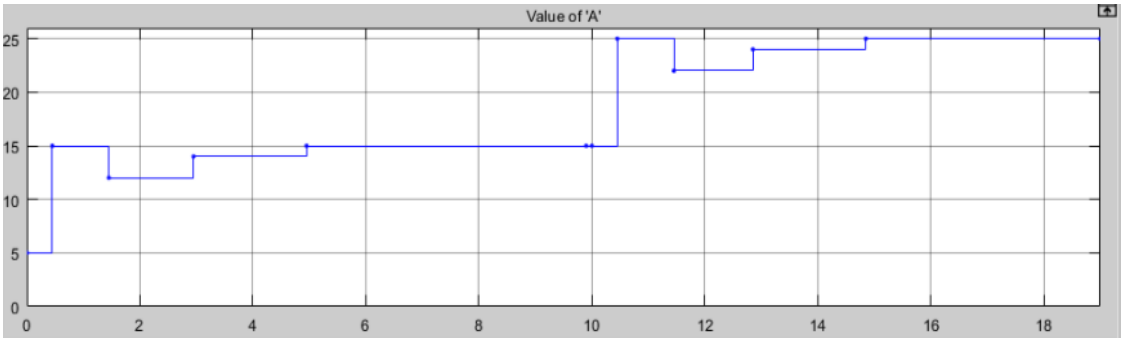
الشكل (٥) قيم الاولوية للمهام مع الزمن الصادرة من الرتل الى المعالج



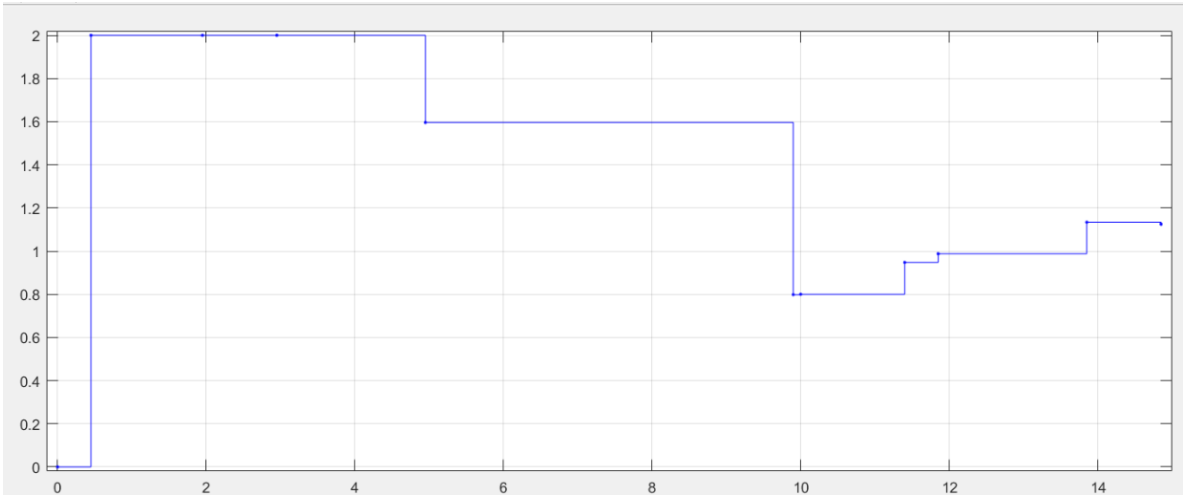
الشكل (٦) معرفات المهام مع الزمن الصادرة من الرتل الى المعالج



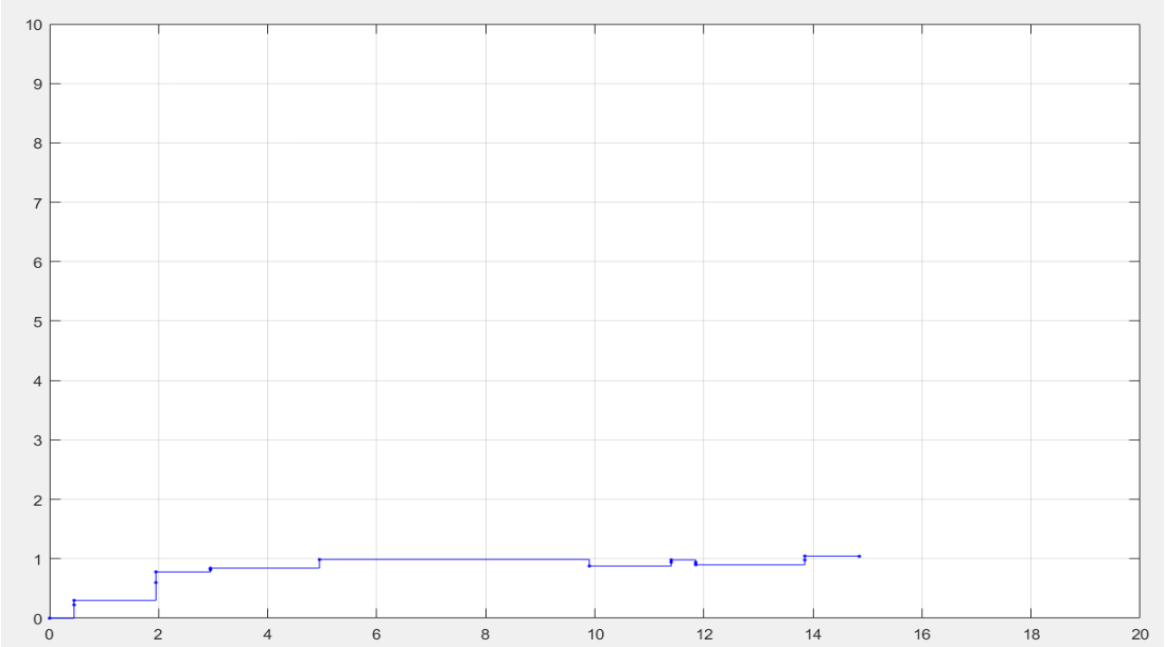
الشكل (٧) عدد المهام الموجودة في الرتل مع الزمن



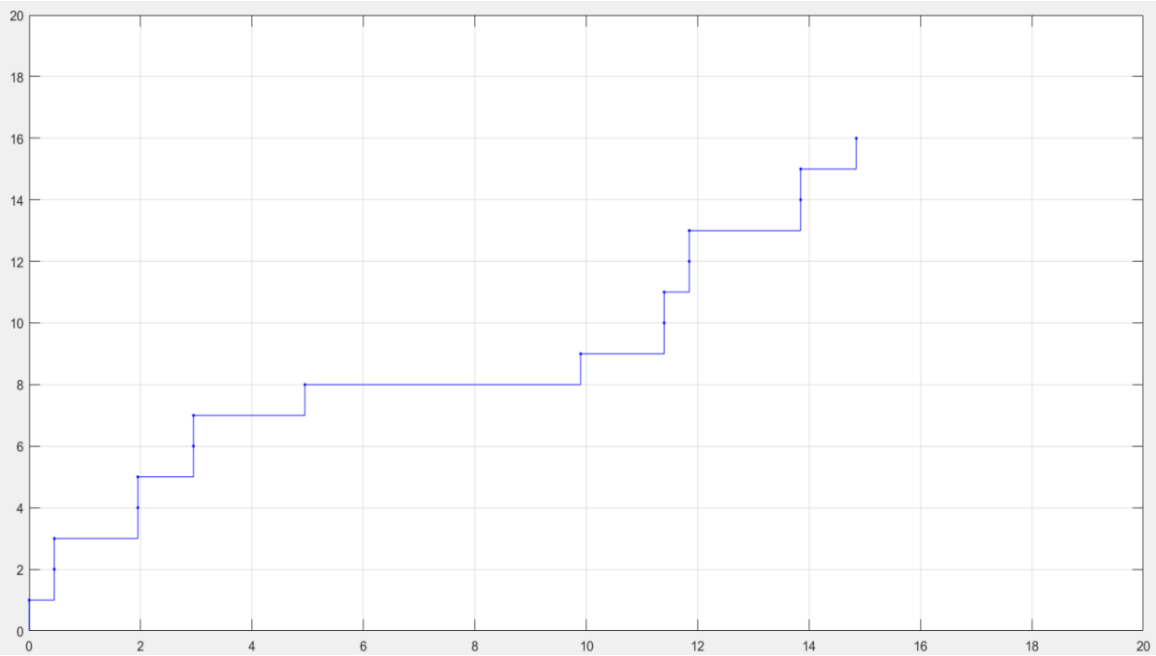
الشكل (٨) تغير قيمة الاشارة A مع الزمن



الشكل (٩) متوسط طول الرتل



الشكل (١٠) متوسط زمن انتظار المهام في الرتل



الشكل (١١) متوسط زمن انتظار المهام في الرتل

٨- الاستنتاجات والتفسيرات :

١. من الشكل (٥) نستنتج ان الرتل يمرر المهام ذو الاولوية الاعلى الى المعالج حيث من هنا ياتي مفهوم الاستباقية حيث عند وجود مهام في الرتل ذات اولوية دنيا وورود مهمة ذات اولوية عليا فانها تمرر فورا ولا تنتظر في الرتل وخصوصا اذا كانت المهمة الواردة هي مهمة المقاطعة ذات المعرف رقم واحد كما يمكننا ملاحظة ان الرتل يكون فارغ (بدون مهام) عند تمرير مهمة الخمول ذات الاولوية الاقل من بين كل المهام السابق ذكرها والتي يتم تولدها وبفواصل زمنية اصغر من الفواصل الزمنية التي تولد بها المهام الاخرى لكي نتمكن بسهولة من التأكد من خلو الرتل من المهام الهامة حيث نلاحظ انه خلال الفترة الزمنية ٥ و ١٠ ثانية يكون التمرير فقط لمهام الخمول .

٢. من الشكل (٦) نستنتج ان الرتل يمكننا من خلال قياس متغير معرفات المهام معرفة تلك المهام التي تم تمريرها من قبل الرتل ومن خلال اقرانه بالاشكال الناتجة عن قياس المتغيرات المختلفة الاخرى يمكننا ومن دون النظر الى بنية الدارة معرفة المهمة ذات معرف معين انها تقابل اولوية معينة ولا بد من ملاحظة المهمة ذات المعرف الصفري والتي تدل على حالة الخمول والتي تقابل الاولوية الادنى .

٣. من الشكل (٧) نستنتج انه يمكننا من خلال قياس متغير عدد المهام معرفة عدد المهام في الرتل في كل لحظة زمنية ففي المثال السابق نجد انه في البداية يكون لدينا مهمتين ذات المعرفين ١ و ٣ ومن ثم تخرج من الرتل المهمة بمعرف ١ ذات الاولوية الاعلى وتبقى في الرتل المهمة ذات الاولوية الادنى ثم تاتي الى الرتل المهمة ذات المعرف ٢ ناتجة عن تفسير ومعالجة المهمة ذات المعرف رقم ١ وبالتالي يكون لدينا في هذه اللحظة اثنان من المهام في الرتل ثم يتم تمرير المهمة بمعرف رقم ٢ وتبقى المهمة ذات المعرف رقم ٣ في الرتل ثم بعد تنفيذ المهمة ذات المعرف ٢ يتم تمرير المهمة ذات المعرف ٣ ويصبح الرتل لا يحوي اي مهام ذات فائدة ثم بعد فترة زمنية ناتجة عن التاخير في المعالج الذي تتعرض له المهمة بمعرف ٣ تاتي الى الرتل المهمة ذات المعرف ٤ ناتجة عن تفسير ومعالجة المهمة ذات المعرف رقم ٣ وبالتالي يكون لدينا في هذه اللحظة واحدة من المهام في الرتل ثم تمرر الى المعالج لتفسر وتعالج وتنفذ العملية الرياضية المطلوبة على الاشارة A ثم يصبح الرتل فارغ من المهام المفيدة حتى الثانية العاشرة ليعاد توليد المهمة بمعرف ١ و المهمة بمعرف ٣ من جديد وتعاد الدورة مرة اخرى بمتوسط طول رتل موافق لما في الشكل (٩) .

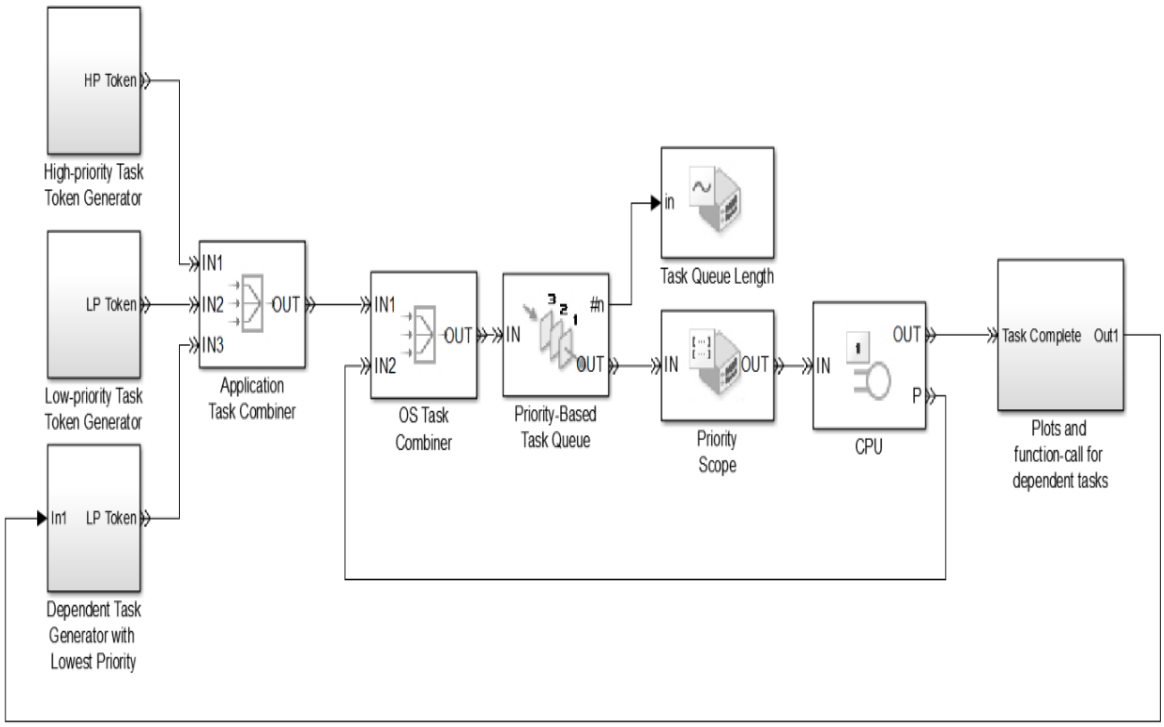
٤. من الشكل (٨) نستنتج انه يمكننا قياس قيمة الاشارة A مع الزمن نتيجة تطبيق العمليات الرياضية الناتجة عن تفسير المهام بعد معالجتها حيث المهمة بمعرف ١ تعالج اولا وتفسر باضافة ١٠ الى قيمة الاشارة المبدئية والتي كانت تحمل القيمة ٥ لتصبح قيمة الاشارة ١٥ ثم تعالج المهمة بمعرف ٢ وتفسر بيتطبيق عملية طرح من قيمة الاشارة بمفدار ٣ لتصبح قيمتها ١٢ ثم بعد معالجة المهمة بمعرف ٣ تصبح قيمة الاشارة ١٤ وبمعالجة المهمة بمعرف ٤ تصبح قيمة الاشارة ١٥ وحتى الثانية العاشرة تبقى الاشارة ثابتة عند القيمة ١٥ لان المهام الخاملة لا يتم تفسيرها بعد معالجتها فرضا وانما تذهب الى المصب بدون تفسير .

٥. من الشكل (١٠) نستنتج انه يمكن قياس متوسط عدد المهام الموجودة في الرتل حيث يتم الحصول على عدد المهام في الرتل ومن ثم ايجاد المتوسط مع الزمن والذي يمكن ان ياخذ قيما عشرية .

٦. من الشكل (١١) نرى انه يمكننا قياس متوسط الزمن الذي ستمكثه المهام في الرتل حتى يتم ترحيلها الى المعالج لمعالجتها .

٣-٩- مثال يركز على العمليات المجدولة من خلال المحاكاة و فهم طبيعة العمل :

يوضح هذا المثال كيفية جدولة المهام وتنفيذها بأولويات مختلفة.

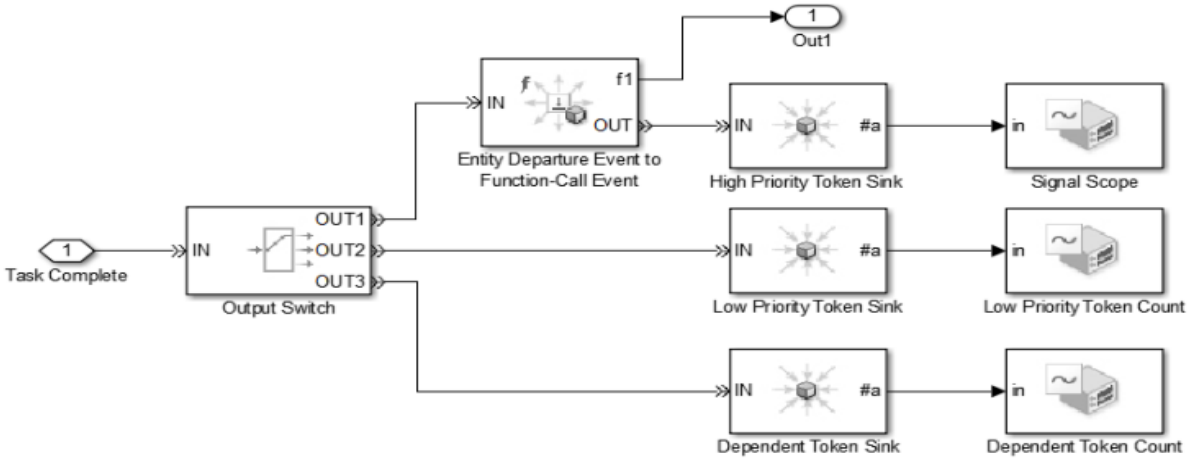


الشكل (١٢) النموذج المعتمد لمحاكاة المهام الاستباقية في النظام

يتم توليد المهمات بمستويين ذوي اولوية باستخدام مولد الكينونات المعتمد على الزمن ثم يتم تعيين سمات محددة لكل من هذه المهمات منها الاولوية و الزمن اللازم لتنفيذ المهمة ويتم ربط كل مهمة برقم عشوائي يسحب من تابع توزع احتمالي كما اسلفنا في المثال السابق مع ملاحظة انه يشير وجود منفذ دخل الى مولد الكينونات إلى أن هذا النوع من المهام يعتمد على إكمال نوع آخر من المهام [7]. يتم فرز المهام وفقًا للأولوية التصاعدية في قائمة انتظار المهام القائمة على الأولوية قبل وحدة المعالجة مباشرةً. ثم يتم تمرير الرموز المميزة إلى وحدة المعالجة لتنفيذ المهام.

هنا اعتمدنا على ثلاث مولدات مهام احداها ذات اولوية عالية ذات قيمة ١ يعتمد على تنفيذها واتمامها مهمة اخرى ذات اولوية اقل ذات قيمة ٣ والاخرى ذات اولوية منخفضة ذات قيمة ٢ .

لا بد لنا من ايضاح محتوى البلوك Plots and function-call for dependent tasks وهو كما مبين في الشكل:

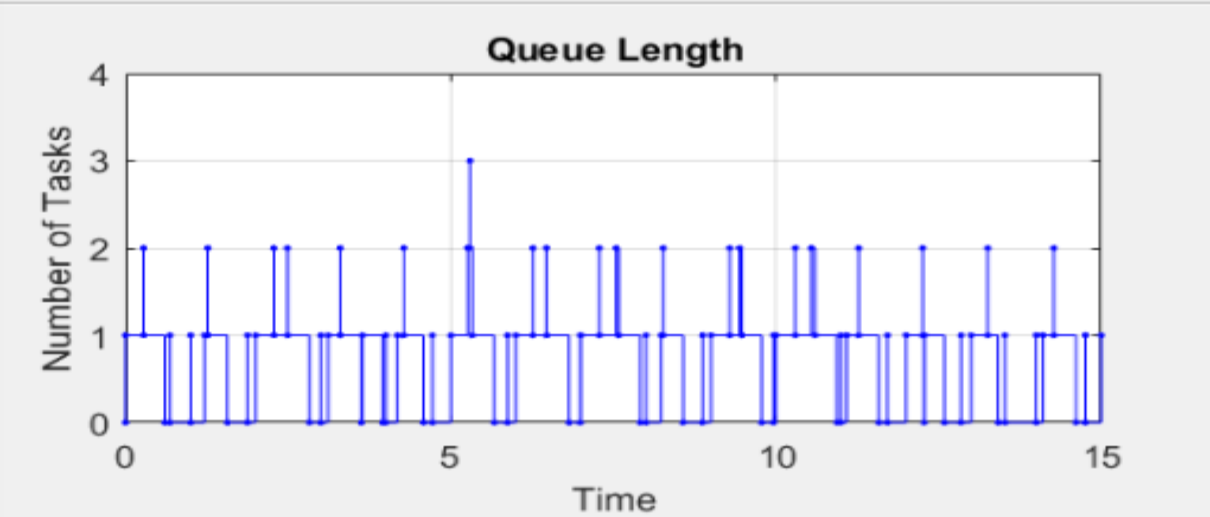


الشكل (١٣) محتوى البلوك Plots and function-call for dependent tasks

فهو يبين انه عند اكمال تنفيذ المهمة ذات الاولوية العليا فانه سوف يتم تفسير المهام بعد معالجتها في المعالج الى توليد المهمة ذات الاولوية التي لها القيمة ٣ والتي تعتمد في توليدها على حدث معالجة المهمة ذات الاولوية التي لها القيمة ١ اما عندما يتم معالجة المهام باولوية ات قيمة ٢ و ٣ فانها سوف تذهب الى المصب دون ان تفسر الى توليد مهمات معينة [8] .

٣-١٠- الاستنتاجات والتفسيرات :

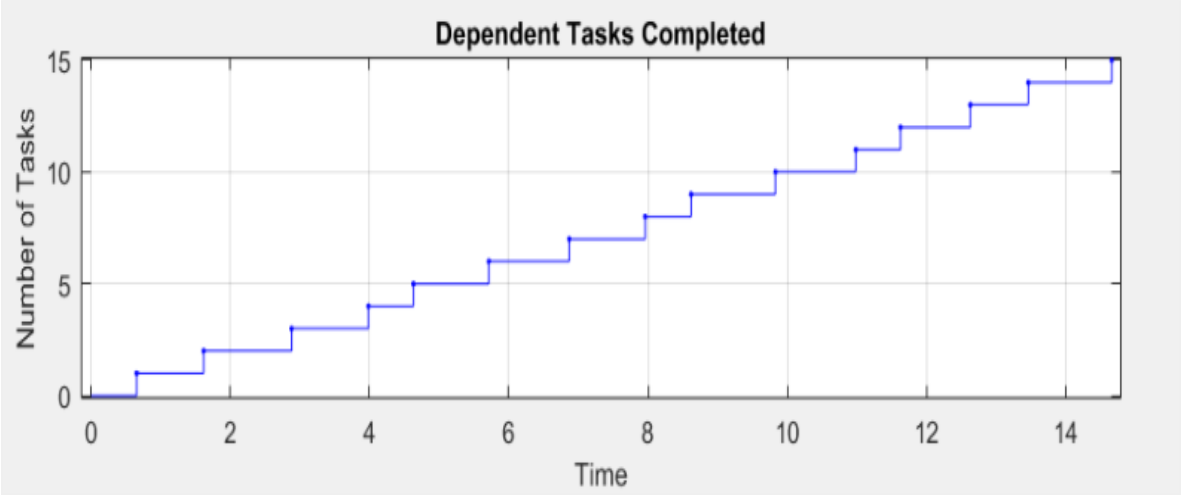
الشكل (١٤) قيمة الاولوية للمهام



الشكل (١٥) طول الرتل

الشكل (١٦) المهام ذات الاولية المنخفضة المنجزة

الشكل (١٧) المهام ذات الاولية المرتفعة المنجزة



الشكل (١٨) المهام التابعة المنجزة

النتائج :

١. من الشكل (١٤) نستنتج انه يمكننا قياس قيمة الأولوية للمهمة التي يتم تنفيذها في وحدة المعالجة واظهارها على شكل محطط يشير إلى أن مهمة ذات أولوية عليا (أولوية ١) تستبق مهمة ذات أولوية منخفضة (الأولوية ٢) وعندما تقوم مهمة ذات أولوية منخفضة باستباق مهمة تابعة (أولوية ٣).

٢. من الشكل (١٥) نستنتج ان طول قائمة الانتظار يزداد في بعض الأحيان عند اكتمال المهمة ذات الأولوية العليا لأن الإكمال يتسبب في إنشاء مهمة تابعة جديدة ذات اولوية دنيا .

٣. من الشكل (١٦) نلاحظ المهام ذات الأولوية المنخفضة المكتملة والتي عددها هو ٢٥ مهمة يتم انجازها خلال فترة المحاكاة التي تقدر ب ١٥ ثانية لان الزمن الفاصل بين توليد مهمتين ذات اولوية ٢ يتراوح بين نصف ثانية و 0.7 من الثانية .

٤. من الشكل (١٧) نستنتج ان عدد المهام ذات الأولوية المرتفعة المكتملة هو ١٥ مهمة يتم انجازها خلال فترة المحاكاة التي تقدر ب ١٥ ثانية لان الزمن الفاصل بين توليد مهمتين ذات اولوية ١ هو ثابت وقيمته ثانية واحدة .

٥. من الشكل (١٨) نستنتج ان عدد المهام التابعة المنجزة يساوي عدد المهام ذات الأولوية العالية التي تم إكمالها نظرًا لأنه يتم إنشاء الرموز المميزة للمهمة التابعة عند اكتمال مهمة ذات أولوية عالية (توليد مهمة تابعة ناتجة عن تفسير مهمة ذات اولوية مرتفعة) .

المراجع :

[1] Copic, Milan, Rainer Leupers, and Gerd Ascheid. "*Reducing idle time in event-triggered software execution via runnable migration and DPM-Aware scheduling*." Integration 70 (2020): 10-20.

[2] Yang, T. C.. Networked control system: a brief survey[J], *Control Theory and Applications. IEE Proceedings*, 2006, 153(4):403B412

[3] Kim, Junhong, and Jaemyung Ahn. "*Task and attitude control scheduling of multiple agile satellites considering task-dependent transition time*." AIAA Scitech 2020 Forum. 2020.

[4] Zhang, Weizhe, Enci Bai, and Jing Li. "*Speeding up the Schedulability Analysis and Priority Assignment of Sporadic Tasks under Uniprocessor FPNS.*" IEEE Transactions on Industrial Informatics (2020).

[5] M. S. Elbamby, M. Bennis, and W. Saad, "*Proactive edge computing in latency-constrained fog networks,*" in 2017 European Conference on Networks and Communications (EuCNC), June 2017, pp. 1–6

[6] Baker T. Stack-Based Scheduling of Real-Time Processes[J].

[7] Sarkar, Chayan, and Marichi Agarwal. "*Systems and methods for scheduling a set of non-preemptive tasks in a multi-robot environment*." U.S. Patent Application No. 16/284,991.

[8] Ning Liu, Huai Liu, Shumin Fei. Optimal Tasks and Messages Scheduling for Asynchronous Networked Control Systems[A]. In: *Proceedings of the IEEE International Conference on Automation and Logistics*, 2007, Jinan, China, 2329-2333