

النظم المضمنة في الزمن الحقيقي

Real Time Embedded Systems

جامعة
المنارة
HARAMA UNIVERSITY

Dr.-Eng. Samer Sulaiman

2020-2021

مفردات المنهاج

- أساسيات النظم المضمنة في الزمن الحقيقي
- عناصر وتقنيات النظم المضمنة في الزمن الحقيقي
- تصميم النظم المضمنة في الزمن الحقيقي



جامعة
المنارة
MANARA UNIVERSITY

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

- المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:
 - نظم تشغيل الزمن الحقيقي:
 - الأسس النظرية لنظم تشغيل الزمن الحقيقي:
 - تتصف معظم نظم الزمن الحقيقي بأنها تشاركية concurrent،
 - تتطلب طبيعة تفاعلها مع الأحداث الخارجية عادة أن تقوم عدة إجراءات بالتخاطب مع العالم الخارجي في نفس الوقت (على التفرع).
 - جدولة الإجراءات:
 - تُعتبر من المهام الأساسية لنظام التشغيل.
 - المتطلبات الضرورية لتحقيق الاحتياجات الزمنية لبرنامج زمن حقيقي
 - خوارزميات جدولة مختلفة عن تلك المستعملة في نظم التشغيل العادية.
 - استراتيجيات خاصة لترتيب استعمال مصادر النظام من قبل إجراءات الزمن الحقيقي حسب أولوياتهم وقيودهم الزمنية.
 - أنواع خوارزميات الجدولة: يوجد نوعان أساسيان لخوارزميات الجدولة في الزمن الحقيقي
 - خوارزميات ما قبل التنفيذ
 - تقوم بجدولة الإجراءات مسبقا بحيث تكون الجدولة الناتجة محققة للقيود الزمنية
 - لا يوجد تضارب في استعمال المصادر (أي لا يستعمل أكثر من إجراء نفس المصدر في نفس اللحظة)،
 - قبل بدء التنفيذ الفعلي للإجراءات. تحاول هذه الخوارزميات أيضا تخفيف كلفة تبديل السياق قدر الإمكان، مما يزيد احتمال إيجاد جدولة محققة لجميع القيود الزمنية.
 - خوارزميات زمن التنفيذ
 - تعطى أولويات ساكنة للإجراءات،
 - تخصص المصادر حسب هذه الأولويات.
 - تعتمد على تقنيات معقدة في زمن التنفيذ لتحقيق الاتصال والتزامن بين الإجراءات
 - الهدف المشترك لخوارزميات الجدولة هو تحقيق الشروط الزمنية لإجراءات الزمن الحقيقي.

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

- المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:
 - نظم تشغيل الزمن الحقيقي:
 - الأسس النظرية لنظم تشغيل الزمن الحقيقي:
 - جدولة الإجراءات:
 - العبء على المعالج وخصائص المهام
 - مجموعة الإجراءات التي يجب على المعالج تنفيذها في لحظة محددة.
 - يمكن لمعالج أن ينفذ إجراءً واحداً على الأكثر في لحظة ما، وكل إجراء يُنفذ على معالج واحد على الأكثر في لحظة ما.
 - يوصف كل إجراء زمن حقيقي τ_i بالواصفات الزمنية التالية:
 - قيود الأسبقية precedence constraints: تحدد فيما إذا كان هناك إجراء (أو إجراءات) يجب أن تُنفذ قبل تنفيذ τ_i
 - زمن الوصول arrival أو التحرير release: رمزه r_{ij} وهو اللحظة التي يحين فيها تنفيذ التكرار رقم j للإجراء τ_i (إذا كان دورياً أي يُكرّر تنفيذه كل دور).
 - الطور phase: رمزه ϕ_i وهو زمن التحرير لأول تكرار للإجراء τ_i .
 - زمن الاستجابة response time: وهو الفترة الزمنية بين لحظة تحرير الإجراء ولحظة اكتمال تنفيذه.
 - الحد الزمني المطلق absolute deadline: رمزه d_i وهو اللحظة الزمنية التي يجب أن ينتهي الإجراء قبلها.
 - الحد الزمني النسبي relative deadline: رمزه D_i وهي القيمة العظمى المسموحة لزمن الاستجابة.
 - نوعية السماح laxity type: هو مقدار الإلحاح urgency في تنفيذ الإجراء τ_i
 - الدور period ورمزه p_i : هو الفترة الزمنية التي تفصل بين لحظة التحرير لإجراء دوري ولحظة التحرير التالية مباشرة.
 - زمن التنفيذ execution time ورمزه e_i : وهو الزمن (الأعظمي) اللازم لإتمام تنفيذ الإجراء τ_i عندما يُنفذ لوحده وتكون جميع المصادر التي يحتاجها متاحة

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

- المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:
 - نظم تشغيل الزمن الحقيقي:
 - الأسس النظرية لنظم تشغيل الزمن الحقيقي:
 - جدولة الإجراءات:
 - نموذج مبسط للإجراء:
- من أجل توصيف بعض خوارزميات الجدولة النظامية المستعملة في نظم الزمن الحقيقي، سنعتمد على نموذج مبسط للإجراء يتضمن الفرضيات التالية:
 - كل الإجراءات الموجودة في النظام دورية
 - الحد الزمني النسبي لكل إجراء يساوي دوره
 - كل الإجراءات مستقلة عن بعضها البعض، أي لا توجد قيود أسبقية بينها
 - لا يحوي أي إجراء قسما غير شُفعي (أي غير قابل للمقاطعة)، وكلفة تبديل السياق مهملة
 - لكل إجراء احتياجات معالجة فقط (أي يحتاج فقط للمعالج) وتُهمل الاحتياجات الأخرى (ذاكرة، دخل/خرج، ...)
- في نظم الزمن الحقيقي يجب أن تكون الجدولة الناتجة عن خوارزمية الجدولة قابلة للتنبؤ predictable،
 - أي أنه يمكننا معرفة الإجراء الذي سيُنقذ لاحقا في كل لحظة.
- تستعمل العديد من نظم تشغيل الزمن الحقيقي خوارزمية الشريط الدوّار round-robin التي تتميز بأنها بسيطة وقابلة للتنبؤ

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• الأسس النظرية لنظم تشغيل الزمن الحقيقي:

• خوارزمية جدولة الشريط الدوار round-robin:

- يجري تنفيذ الإجراءات تسلسلياً حتى تنتهي وغالباً ما توجد في النظم ذات التنفيذ الدوري cyclic executive
- يُخصَّص لكل إجراء شريحة (حصّة) زمنية ثابتة quantum or slice يُنفَّذ خلالها.
- تُستعمل نبضات ساعة ذات دور ثابت يساوي الحصّة الزمنية لمقاطعة النظام دورياً
- يستمر تنفيذ الإجراء حتى ينتهي تنفيذه أو تنتهي حصته الزمنية بحدوث مقاطعة نبضات الساعة.
- إذا لم ينته تنفيذ الإجراء، يُحفظ سياقه ويوضع في نهاية رتل الإجراءات الجاهزة.
- ليقوم النظام باسترجاع سياق الإجراء التالي الموجود في بداية الرتل ويتابع تنفيذه.
- تتميز بأنها جدولة عادلة غير شُفعية للإجراءات التي لها نفس الأولوية بتوزيع زمن المعالج عليها بالتساوي
- يمكن إدخال مفهوم الأولوية الشُفعية لهذه الخوارزمية وذلك بالسماح للإجراءات ذات الأولوية الأعلى بشُفع المعالج إذا كان ينفذ إجراءً أقل أولوية لحظة وصولها.

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• الأسس النظرية لنظم تشغيل الزمن الحقيقي:

• خوارزمية جدولة الشريط الدوار round-robin:

• مثال: ليكن لدينا نظام يقوم بتنفيذ الاجرائيات كما هو مبين بالشكل.

• الاجراءان A و C لهما نفس الأولوية والى B له أولوية أعلى.

• يبدأ A بالتنفيذ لبعض الوقت، إلى أن يصل الإجراء B

• ذو الأولوية الأعلى ويقوم بشفع A ويستمر بالتنفيذ حتى انتهائه.

• بعدها يعود A للتنفيذ إلى أن تنتهي حصته الزمنية

• ليتم بعدها الانتقال لتنفيذ الإجراء C حتى انتهاء حصته الزمنية وهكذا.

• التنفيذ الدوري cyclic executive:

• يُستعمل بكثرة في العديد من نظم الزمن الحقيقي لأنه بسيط ويولد جدولة قابلة للتنبؤ.

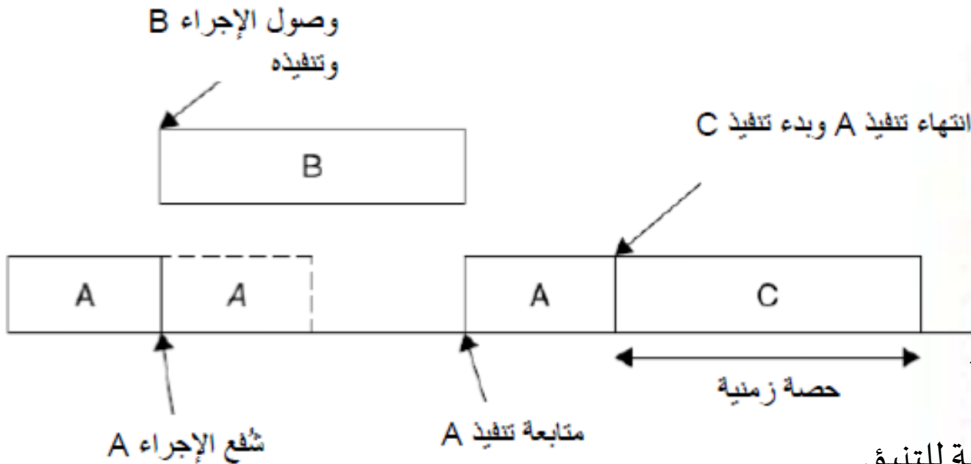
• يقوم بسلسلة ومراكبة تنفيذ إجراءات دورية حسب جدول محدد مسبقاً قبل التنفيذ.

• يتخذ هذا الأسلوب قرارات الجدولة دورياً ومسبقاً وليس في أية لحظة مثل الشريط الدوار.

• تسمى الفترات الزمنية (المتساوية) التي تفصل بين لحظات اتخاذ قرارات الجدولة بالأطر frames أو الأدوار الصغيرة minor cycles و يُرمز ل طولها بالرمز f

• يسمى أيضاً "حجم الإطار"

• يمثل الدور الكبير major cycle or hyper period المضاعف المشترك الأصغر لجميع أدوار إجراءات النظام.



المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• الأسس النظرية لنظم تشغيل الزمن الحقيقي:

• التنفيذ الدوري cyclic executive:

• بما أن قرارات الجدولة تؤخذ فقط في بداية كل إطار، لذلك لا يحدث أي شُفع داخل الإطار.

• إن طور phase أي إجراء دوري هو مضاعف صحيح غير سالب من حجم الإطار.

• بعض القيود المفروضة على حجم الإطار f :

• يجب أن يكون الإطار كبيراً كفاية بحيث يبدأ وينتهي أي إجراء ضمن إطار واحد.

• أي يجب أن يكون حجم الإطار f أكبر من كل أزمان التنفيذ e_i للإجراءات T_i وبالتالي: $C_1 : f \geq \max_{1 \leq i \leq n} (e_i)$

• من أجل طول الجدول الدوري أصغر ما يمكن، يجب أن يحتوي الدور الكبير على عدد صحيح من الأطر: $C_2 : [p_i/f] - p_i/f = 0$

• وللتأكد من أن كل إجراء ينتهي قبل حده الزمني، يجب أن يكون الإطار صغيراً بحيث يكون هناك إطاراً واحداً على الأقل بين كل تحرير لإجراء وبين حده الزمني.

• يمكن إعطاء العلاقة المشتقة من الحالة الأسوأ التي تحدث عندما يبدأ دور إجراء ما مباشرة بعد بدء الإطار، ولا يمكن بالتالي

تحريره قبل بدء الإطار التالي: $C_3 : 2f - \gcd(p_i, f) \leq D_i$

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• الأسس النظرية لنظم تشغيل الزمن الحقيقي:

• التنفيذ الدوري cyclic executive:

• بعض القيود المفروضة على حجم الإطار f:

• مثال: ليكن لدينا نظاما يحتوي على الإجراءات الدورية المبينة في الجدول التالي:

- قيمة الدور الكبير لهذه الإجراءات هي 660 (وهو المضاعف المشترك الأصغر للقيم 15 و 20 و 22)

• نطبق الشروط الثلاثة السابقة C_1 و C_2 و C_3 فنحصل على العلاقات التالية:

$$C_1 : \forall i f \geq e_i \Rightarrow f \geq 3$$

$$C_2 : \lfloor p_i/f \rfloor - p_i/f = 0 \Rightarrow f = 2, 3, 4, 5, 10, \dots$$

$$C_3 : 2f - \gcd(p_i, f) \leq D_i \Rightarrow f = 2, 3, 4, 5$$

- نستنتج مما سبق أن القيم الممكنة لحجم الإطار f هي إما 3 أو 4 أو 5

τ_i	p_i	e_i	D_i
τ_1	15	1	14
τ_2	20	2	26
τ_3	22	3	22

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• الأسس النظرية لنظم تشغيل الزمن الحقيقي:

• جدولة الأولويات الثابتة – الخوارزمية "الترتبة المعدل" (Rate Monotonic (RM):

• في النظم ذات الأولويات الثابتة، يُعطى لكل إجراء أولوية محددة نسبةً لباقي الإجراءات.

• تُعتبر من أهم خوارزميات الجدولة في هذه النظم،

• تعطي جدولة أمثليه في نموذج الإجراءات المبسط

• يُعطى لكل إجراء أولوية تتناسب عكسا مع دوره:

• فأولوية الإجراء الذي له دور كبير أصغر من أولوية الإجراء الذي له دور قصير.

• مثال: ليكن لدينا مجموعة الإجراءات التالية الموضحة في الجدول:

• نفترض أن زمن تحرير كل الإجراءات الثلاثة هو 0.

• بما أن الإجراء τ_1 له أصغر دور، فإن أولويته أعلى من أولويات باقي الإجراءات، لذلك يُنفَّذ أولاً.

• في اللحظة 4 يصل التكرار الثاني للإجراء τ_1 الذي يقوم بشُفع الإجراء τ_3 لأن أولويته أعلى

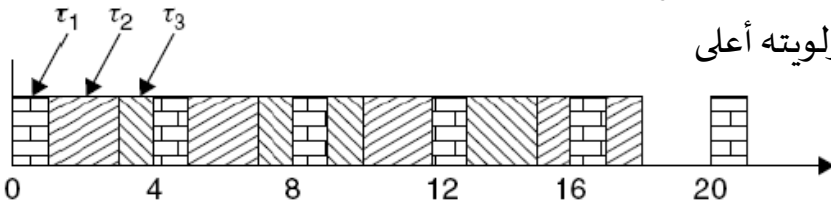
• تسمى القيمة u_i "الانشغالية" utilization وهي النسبة (من واحد) التي

يُشغل فيها إجراء له دور p_i وزمن تنفيذ e_i المعالج.

• انشغالية المعالج في حالة n إجراء تعطى بالعلاقة

$$U = \sum_{i=1}^n e_i/p_i$$

τ_i	e_i	p_i	$u_i = e_i/p_i$
τ_1	1	4	0.25
τ_2	2	5	0.4
τ_3	5	20	0.25



المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• الأسس النظرية لنظم تشغيل الزمن الحقيقي:

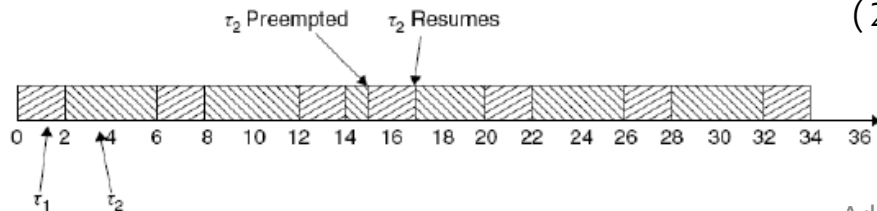
• جدولة الأولويات الديناميكية – خوارزمية "الحد الأبعد أولاً" (Earliest-Deadline-First (EDF):

- تتغير أولويات الإجراءات أثناء عمل النظام في نظم الأولويات الديناميكية
- تعتبر من أشهر خوارزميات هذه النظم، حيث تعتمد أولويات الإجراءات على حدودها الزمنية بدلاً من أزمان تنفيذها.
- في أية لحظة، الإجراء الذي له أقرب حد زمني هو الإجراء ذو الأولوية الأعلى.
- في نظام زمن حقيقي يحتوي على n إجراء دوري حدودها النسبية تساوي دورها
- الشرط اللازم ليكون قابلاً للجدولة حسب هذه الخوارزمية
- مثال: ليكن لدينا مجموعة الإجراءات التالية الموضحة في الجدول:

τ_i	p_i	e_i
τ_1	5	2
τ_2	7	4

$$\sum_{i=1}^n \left(\frac{e_i}{p_i} \right) \leq 1$$

- بفرض لحظة تحرير كلا الإجراءين هي اللحظة 0،
- يبدأ الإجراء τ_1 بالتنفيذ لأن له حد زمني أقرب.
- في اللحظة $t=2$ ينتهي تنفيذ τ_1 ويبدأ تنفيذ τ_2
- في اللحظة $t=5$ يصل التكرار الثاني للإجراء τ_1 لكنه لا يقوم بشُفع τ_2 لأن حده الزمني أبعد من حد τ_2
- في اللحظة $t=15$ يقوم τ_1 بشُفع τ_2 لأن حده الزمني (وهو 20) أقرب من الحد الزمني للإجراء τ_2 (وهو 21).
- يتابع τ_2 تنفيذه بعد انتهاء τ_1 في اللحظة $t=17$



المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

- المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:
 - نظم تشغيل الزمن الحقيقي:
 - مزامنة الإجراءات ومشاركة المصادر:
 - افترضنا حتى الآن أن إجراءات النظام مستقلة ويمكن أن يجري شُفعها في أية لحظة.
 - هذه الفرضيات غير واقعية عمليا في معظم النظم، إذ تحتاج الإجراءات أن تتصل وتتفاعل مع بعضها البعض
 - المصادر Resources:
 - المصادر هي أية تسهيلات يقدمها النظام للإجراءات لمساعدتها في تأدية عملها.
 - بعض الأمثلة على المصادر المشتركة في النظام:
 - الملفات
 - التجهيزات المختلفة (طابعات، شاشات، ...)
 - المتحولات المشتركة في الذاكرة التي غالبا ما تُستعمل للاتصال بين الإجراءات
 - هناك الكثير من التقنيات البرمجية المستعملة كمتحولات مشتركة في الذاكرة لتسهيل الاتصال بين الإجراءات نذكر منها:
 - Buffer بأنواعه المتعددة
 - صندوق البريد mail box
 - الأرتال queues

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• الأقسام الحرجة critical section:

• مهما كان المصدر الذي يحتاج للإجراء للوصول إليه، عادةً ما يحتاج أن يصل إليه وحيداً ودون مقاطعة، وإلا يمكن أن تحدث حالات خاصة تجعل نتيجة التنفيذ غير صحيحة.

τ_1	τ_2
(1) $x = i;$	(4) $y = i;$
(2) $x = x + 1;$	(5) $y = y - 1;$
(3) $i = x;$	(6) $i = y;$

• مثال بفرض لدينا الإجراءين التاليين τ_1 و τ_2 الموضحان بالجدول التالي:

• المتحول x محلي في الإجراء τ_1 والمتحول y محلي في الإجراء τ_2

• المتحول i هو متحول مشترك بين الإجراءين قيمته الابتدائية

صفر وهو المصدر المشترك

• إذا فرضنا أن كلاً من الإجراءين ينقذ مرة واحدة،

• تكون القيمة النهائية الصحيحة في i هي الصفر.

• لكن لو فرضنا أن τ_1 بدأ بالتنفيذ وجرى شُفعه بين التعليمتين (1) و (2)، وانتقل التنفيذ للإجراء τ_2 ، ثم عاد التنفيذ للإجراء τ_1

بعد الانتهاء من التنفيذ الكامل للإجراء τ_2

• عندها تكون القيمة النهائية في i تساوي 1، وهي قيمة خاطئة.

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

- المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:
 - نظم تشغيل الزمن الحقيقي:
 - مزامنة الإجراءات ومشاركة المصادر:
 - الأقسام الحرجة critical section:
 - تسمى مجموعة التعليمات التي يتعامل الإجراء فيها مع المصدر المشترك "القسم الحرج" critical section
 - يمكن أن يحتوي إجراء ما على أقسام حرجة لنفس المصدر أو لمصادر مختلفة.
 - إذا دخل إجراءان قسمين حرجين لنفس المصدر وفي نفس الوقت، يؤدي ذلك لحدوث أخطاء في نتائج التنفيذ، مما يمكن أن يؤدي لنتائج كارثية في بعض الحالات.
 - هناك حاجة كبيرة لتقنيات تمنع حدوث مثل هذا الوضع في معظم نظم الزمن الحقيقي العملية.
 - من هذه التقنيات نذكر "العلامة" semaphore و"المراقب" monitor وتعليمة "اختبر وضع" test and set
 - العلامات semaphores:
 - موقع في الذاكرة يعمل كقفل يحمي القسم الحرج المتعلق بمصدر ما.
 - عادة ما تكون قيمة ثنائية binary (صفرًا أو واحدًا)
 - عندما تكون $S=1$ ، يدل ذلك على أن القفل مفتوح ويمكن لأي إجراء الدخول للقسم الحرج.
 - بينما عندما $S=0$ ، يدل ذلك على أن القفل مغلق ويوجد حالياً إجراء واحد يعمل في القسم الحرج،
 - يجب على جميع الإجراءات التي تريد الدخول إليه الانتظار ريثما يخرج ذلك الإجراء منه.
 - يُعرّف على العلامة عمليتان أساسيتان هما $P(S)$ (إنتظار وإغلاق) و $V(S)$ (تحرير).

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• العلامات semaphores:

- موقع في الذاكرة يعمل كقفل يحيي القسم الحرج المتعلق بمصدر ما.
- عادة ما تكون قيمة ثنائية binary (صفرًا أو واحدًا)
- عندما تكون $S=1$ ، يدل ذلك على أن القفل مفتوح ويمكن لأي إجراء الدخول للقسم الحرج.
- بينما عندما $S=0$ ، يدل ذلك على أن القفل مغلق ويوجد حالياً إجراء واحد يعمل في القسم الحرج،
- يجب على جميع الإجراءات التي تريد الدخول إليه الانتظار ريثما يخرج ذلك الإجراء منه.
- يُعرّف على العلام عملتان أساسيتان هما $P(S)$ (إنتظار وإغلاق) و $V(S)$ (تحرير).
- نحتاج لعلام مستقل لحماية كل مصدر مشترك.
- إذا احتاج قسم حرج ما للوصول إلى عدة مصادر مشتركة، عندها يجب استدعاء العملية P من أجل كل العلامات الموافقة لهذه المصادر قبل الدخول إليه لحجزها كلها.
- غالباً ما تكون العمليتان P و V جزءاً من نواة نظام التشغيل أو استدعاءات لخدمات النظام، لأنها يجب أن تُنفَّذ دون مقاطعة لكي تعمل بشكل صحيح
- يجب استعمال العلامات بحذر وفهم عملها جيداً، وإلا أدى سوء استعمالها لحدوث مشاكل منطقية مثل "المنع المتبادل" deadlock

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• المنع المتبادل Deadlock:

• يمكن أن تحدث هذه المشكلة الخطيرة عندما تحاول عدة إجراءات الوصول إلى مصدرين على الأقل، مما يؤدي لتوقفها كلها إذا جرى تنفيذها

• مثال: بفرض أنه لدينا إجرائين TASK_A و TASK_B

• تحتاج كلاهما الوصول إلى مصدرين يحميهما العلامان S و R

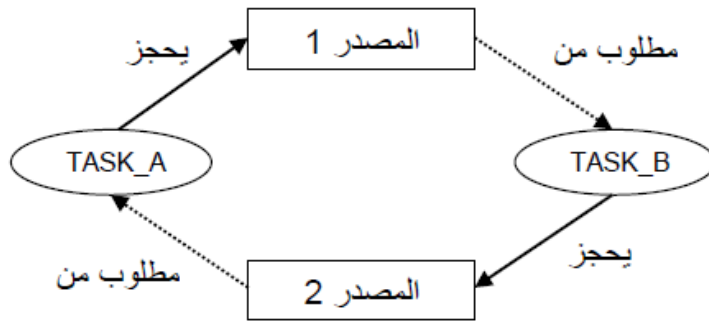
• يحجز الإجراء TASK_A المصدر الأول بعد استدعاء P(S) ويبدأ باستعماله

• قبل أن يصل إلى الاستدعاء P(R) تحدث مقاطعة وينتقل التنفيذ إلى TASK_B الذي يحجز المصدر الثاني ويستمعه.

• عندما يصل التنفيذ إلى P(S) يتوقف TASK_B عن التنفيذ لأن الإجراء TASK_A يحجز المصدر الأول.

• عندها ينتقل التنفيذ إلى الإجراء TASK_A الذي يصل بدوره للاستدعاء P(R) ويتوقف لأن TASK_B يحجز المصدر الثاني.

• بهذا يحجز كل إجراء مصدر وينتظر المصدر الذي يحجزه الإجراء الآخر إلى ما لا نهاية



المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• المنع المتبادل Deadlock:

- إن المنع المتبادل مشكلة خطيرة جداً لأنها تحدث باحتمال صغير جداً ومن الصعب معرفة أنها موجودة بأي اختبار إلى أن تحدث.
- كما أن حلولها ليست سهلة ولها نتائج عديدة غير مرغوبة، من أهمها إضافة عبء حسابي على النظام يؤدي لإضعاف أدائه.
- طريقة لتفادي حدوث المنع المتبادل:

• خوارزمية المصرفي the banker's algorithm:

- تُستعمل لتفادي الحالات الغير آمنة التي يمكن أن تؤدي لحدوث المنع المتبادل.
- اقترحها Dijkstra عام 1968 وهي مشابهة لعمل المصارف في المدن الصغيرة من حيث الحفاظ على مخزون من السيولة النقدية على سبيل الاحتياط. يودع الناس أموالهم في المصرف ويمكنهم سحبها في أية لحظة.
- لا يحتفظ المصرف بكل الأموال السائلة في حوزته، بل يستثمر 95% منها ويحتفظ بنسبة 5% كسيولة نقدية احتياط في حال أراد بعض المودعين سحب جزءاً من إيداعاتهم.
- إذا طلب العديد من الزبائن سحب إيداعاتهم، عندها لن يتمكن المصرف من تلبية جميع الطلبات.
- وبشكل مشابه تفترض خوارزمية المصرفي وجود عدة مصادر من نوع واحد فقط، لكن يمكن تمديدها بحيث تعمل مع عدة أنواع من مصادر
- تتأكد الخوارزمية في كل لحظة من أن عدد المصادر المحجوزة من قِبَل إجراءات النظام زائد عدد المصادر التي يحتاج إليها أحد هذه الإجراءات لإتمام تنفيذه لا يتجاوز العدد الكلي المتاح من المصدر.
- نقول في هذه الحالة أن النظام في حالة آمنة.
- وبالعكس، يكون النظام في حالة غير آمنة إذا كان المجموع السابق أكبر تماماً من العدد الكلي المتاح من المصدر.
- من الممكن لكن ليس من الضروري أن تؤدي الحالات الغير آمنة إلى منع متبادل، لكن من المؤكد أن الحالات الآمنة لن تؤدي أبداً لحدوث منع متبادل. لذلك تضمن الخوارزمية بقاء النظام في حالة آمنة وتضمن بالتالي عدم حدوث المنع المتبادل

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• المنع المتبادل Deadlock:

• طريقة لتفادي حدوث المنع المتبادل:

• خوارزمية المصرفي the banker's algorithm:

• مثال: بفرض نظاماً مكوناً من ثلاثة إجراءات A و B و C و 10 مصادر من نوع واحد (كتل ذاكرة مثلاً)

• إذا علمنا أن الإجراءات A لن يحتاج لحجز أكثر من 6 مصادر في نفس الوقت

• لن يحتاج B لأكثر من 5

• لن يحتاج C بأكثر من 7

• يحتاج النظام لبناء جدول

في كل لحظة لتتبع المصادر

المتاحة والمطلوبة في تلك

اللحظة.

العدد الذي من الممكن أن يحتاج إليه	عدد المصادر المحجوزة حالياً	العدد الأعظمي المطلوب	الإجراء
6	0	6	A
5	0	5	B
7	0	7	C
10	العدد الكلي المتاح حالياً		

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• المنع المتبادل Deadlock:

• طريقة لتفادي حدوث المنع المتبادل:

• خوارزمية المصرفي the banker's algorithm:

• مثال:

• عند كل طلب حجز للمصادر، يحدّث نظام التشغيل هذا الجدول للتأكد من أن النظام ما زال في وضع آمن.

العدد الذي من الممكن أن يحتاج إليه	عدد المصادر المحجوزة حالياً	العدد الأعظمي المطلوب	الإجراء
4	2	6	A
2	3	5	B
6	1	7	C
4	العدد الكلي المتاح حالياً		

العدد الذي من الممكن أن يحتاج إليه	عدد المصادر المحجوزة حالياً	العدد الأعظمي المطلوب	الإجراء
2	4	6	A
2	3	5	B
5	2	7	C
1	العدد الكلي المتاح حالياً		

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• انعكاس الأولوية priority inversion:

- من أهم نتائج استعمال تقنيات مزامنة الإجراءات كالعلاّمت وغيرها في نظم الزمن الحقيقي حدوث انعكاس لأولوية إجراءات الزمن الحقيقي
- يُنفَّذ الإجراء ذو الأولوية المنخفضة عوضاً عن الإجراء ذي الأولوية المرتفعة فقط لأنه يحجز مصدراً يحتاج إليه ذلك الإجراء.
- هذا الأمر غير مرغوب فيه نهائياً لأنه يمكن أن يؤدي لأن يتعدى إجراء حساس جداً حده الزمني بسبب إجراء ذي أولوية منخفضة.

• مثال:

• بفرض لدينا نظام يحتوي على ثلاثة إجراءات τ_1 و τ_2 و τ_3 .

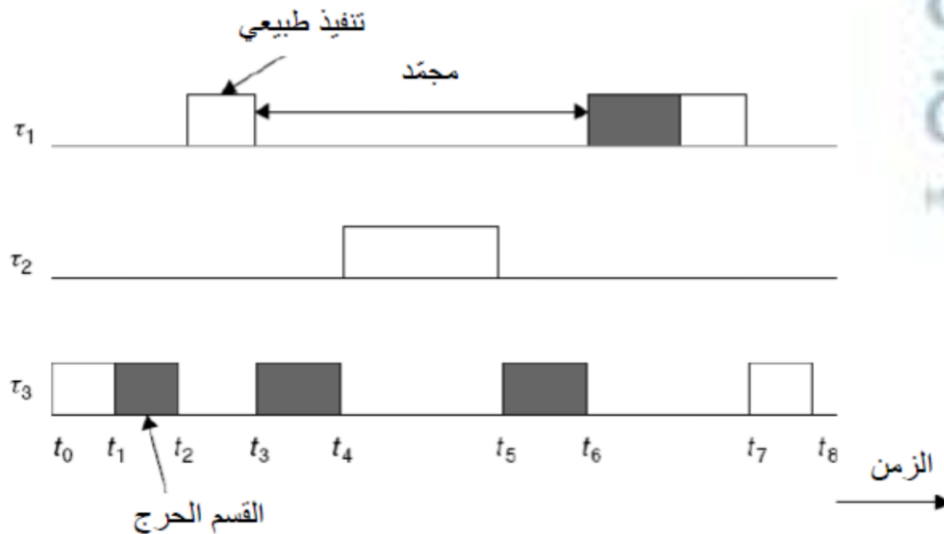
• أولوية τ_1 أعلى من أولوية τ_2

• أولوية τ_2 أعلى من أولوية τ_3 .

• يتشارك τ_1 و τ_3 بمصدر مشترك ويحتاجان لتقنية مزامنة تنظم وصولهما إليه (علام مثلًا مع عمليات P و V الخاصة به).

• لا يحتاج τ_2 للوصول إلى هذا المصدر.

• لنفترض أنه حدث تنفيذ الإجراءات حسب المخطط الزمني التالي (القسم الحرج ملون باللون الغامق)



المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• انعكاس الأولوية priority inversion:

• مثال:

- يبدأ $T3$ العمل في اللحظة $t0$ ويقوم بقفل العلام في اللحظة $t1$ لأنه يريد العمل مع المصدر المشترك
- يصل الإجراء $T1$ في اللحظة $t2$ ويقوم بشُفع الإجراء $T3$ قبل أن ينتهي من القسم الحرج
- لأن له أولوية أعلى. في اللحظة $t3$ يحاول الإجراء $T1$ قفل العلام لأنه يريد أيضا العمل مع المصدر المشترك، لكنه يدخل في حالة انتظار (يصبح مجمّداً) لأن العلام محجوز مسبقاً من قبل $T3$ عندها، يعود التنفيذ إلى $T3$ ليتابع قسمه الحرج
- في اللحظة $t4$ يصل الإجراء $T2$ الذي له أولوية أعلى من $T3$ فيقوم بشُفعه
- بما أن $T2$ لا يحتاج للوصول إلى المصدر المشترك، يستمر تنفيذه إلى أن ينتهي في اللحظة $t5$ مطيلاً بذلك زمن انتظار $T1$ ذي الأولوية الأعلى بمقدار زمن تنفيذ $T2$
- يعود بعدها التنفيذ للإجراء $T3$ إلى أن ينتهي القسم الحرج في اللحظة $t6$ ويحررّ العلام
- عندها ينتقل التنفيذ للإجراء $T1$ لأن أولويته أعلى ولأن العلام الذي ينتظره أصبح حراً
- بعد أن ينتهي تنفيذ الإجراء $T1$ في اللحظة $t7$ يعود التنفيذ للإجراء $T3$ حتى ينتهي تنفيذه في اللحظة $t8$
- من غير المرغوب به أن يحصل انعكاس للأولوية في نظم الزمن الحقيقي. لذلك، تُستعمل في هذه الحالة خوارزميات جدولة مختلفة عن الخوارزميات السابقة بهدف التقليل قدر الإمكان من هذه الظاهرة.
- طرق الحل:
 - خوارزمية "ورثة الأولوية"
 - خوارزمية "سقف الأولوية"

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

- المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:
 - نظم تشغيل الزمن الحقيقي:
 - مزامنة الإجراءات ومشاركة المصادر:
 - انعكاس الأولوية priority inversion:
 - طرق الحل:
 - خوارزمية "وراثة الأولوية" priority inheritance:
 - تتغير أولويات الإجراءات ديناميكياً بحيث يأخذ الإجراء الذي يعمل ضمن القسم الحرج الأولوية العظمى للإجراءات التي تنتظر خروجه من القسم الحرج.
 - عندما يجمّد إجراء ما T_i إجراءات أخرى ذات أولوية أعلى، يرث مؤقتاً أعلى أولوية من أولويات هذه الإجراءات إلى أن يخرج من القسم الحرج حيث يستعيد أولويته الأصلية.
 - الخطوات الأساسية للخوارزمية:
 - سيدخل أي إجراء مهما كانت أولويته في حالة انتظار إذا حاول حجز علامة محجوز من قبل إجراء آخر.
 - إذا جُمّد إجراء T_1 من قبل إجراء آخر T_2 وكانت أولوية T_1 أعلى من أولوية T_2
 - يرث T_2 أولوية T_1 طالما بقي مجمداً له، ويعمل وفق هذه الأولوية الجديدة الموروثة
 - عندما يخرج T_2 من القسم الحرج الذي سبب هذا التجميد، تعود أولويته إلى القيمة التي كانت عليها قبل دخول القسم الحرج
 - وراثة الأولوية متعددة.
 - إذا كان T_3 يجمّد T_2 و T_2 يجمّد T_1 وأولوية T_3 أقل من أولوية T_2 وأولوية T_2 أقل من أولوية T_1
 - يرث T_3 أولوية T_1 عبر T_2

المكونات الأساسية للنظم المضمنة في الزمن الحقيقي

• المكونات البرمجية للأنظمة المضمنة في الزمن الحقيقي:

• نظم تشغيل الزمن الحقيقي:

• مزامنة الإجراءات ومشاركة المصادر:

• انعكاس الأولوية priority inversion:

• طرق الحل:

• خوارزمية "سقف الأولوية" priority ceiling:

• هي تحسين لخوارزمية وراثية الأولوية، بحيث يُمنع أي إجراء من الدخول إلى القسم الحرج إذا كان هذا سيؤدي إلى تجميده.

• يُعطى لكل مصدر أولوية (تسمى سقف الأولوية) تساوي أعلى أولوية لإجراء سيستعمل هذا المصدر

• الفرق بين هذه الخوارزمية والخوارزمية السابقة هو أنه يمكن لإجراء τ_i أن يُمنع من دخول القسم الحرج إذا وجد علامة محجوز من قبل إجراء آخر وله سقف أولوية أكبر أو يساوي أولوية τ_i

• مثال: ليكن لدينا نظام مؤلف من ثلاثة إجراءات 1τ و 2τ و 3τ ويوجد فيه المصادر التالية

• إذا فرضنا أن 2τ يحجز حالياً المصدر S_2 ، وبدأ الإجراء 1τ بالعمل وطلب حجز المصدر S_1 ، عندها سيجمد 1τ ولن يحجز المصدر S_1 رغم أنه حر وذلك لأن أولويته ليست أكبر تماماً من سقف أولوية المصدر المحجوز S_2

القسم الحرج	الإجراءات التي تصل إليه	سقف الأولوية
S_1	τ_1, τ_2	$P(\tau_1)$
S_2	τ_1, τ_2, τ_3	$P(\tau_1)$
S_3	τ_3	$P(\tau_3)$
S_4	τ_2, τ_3	$P(\tau_2)$