

أنظمة الإدخال والإخراج (I/O system)

نظام المدخلات والمخرجات

يقصد بعمليات الإدخال والإخراج تبادل المعلومات بين وحدات الإدخال والإخراج ومسجلات وحدة المعالجة المركزية أو بين وحدات الإدخال والإخراج والذاكرة الرئيسية.

ومن الطرق التي تنفذ عمليات الإدخال والإخراج هي:

1. استخدام طريقة المسح بحيث لكل وحدة إدخال أو إخراج مفتاح معين يميزه (صفر أو واحد) يحدد فيها المبرمج منافذ الإدخال والإخراج المراد استخدامها وذلك من خلال تعليمات مباشرة.
2. الإدخال والإخراج باستخدام مفهوم الاعتراض **interrupt** (الاعتراض هو حدث استثنائي يضطر نظام التشغيل عند الاستجابة له وقف تنفيذ العمل الخاضع للتنفيذ من أجل تنفيذ عمل جديد يسمى العمل المعترض , وتمتلك نظم التشغيل برمجيات خاصة لمعالجة الاعتراض تسمى بمعالج الاعتراض (**interrupt handler**)).

وفيما يلي عناوين المواضيع التي سنتكلم عنها بشئ من الإيجاز لتوضيحها بإذن الله...

1. إن الكمبيوتر يعتمد على المدخلات والمخرجات مع الإجراءات التي تتم بداخله.
2. أجهزة الإدخال والإخراج.
3. تطبيقات المدخلات والمخرجات.
4. تحويل متطلبات المدخلات والمخرجات إلى عمليات على الأجهزة.
5. الأداء.

أولاً: إن الكمبيوتر يعتمد على المدخلات والمخرجات مع الإجراءات التي تتم بداخله:
أ. الوظائف المهمة مثل: تصفح الويب أو تحرير مـــــــف.

- ب. أن نظام التشغيل يدير أجهزة الإدخال والإخراج والعمليات المختلفة.
- ج . أن هناك طائفة واسعة من آلات المدخلات والمخرجات.
- د . البرامج والأجهزة التي تكون الكمبيوتر.

ثانيا: أجهزة الإدخال والإخراج:

هناك أشكال متعددة وأنواع مختلفة للأجهزة المستخدمة في الإدخال والإخراج:

- (Port - a connection point) المنافذ ونقاط الاتصال.
 - (Controller) الأجهزة المتحكممة والمحولات.
 - الأجهزة الناقلة وهي نوعين:
- أ. (Memory-mapped I/O) الذاكرة التي تستخدم لتخزين الأشياء المراد نقلها.
- ب. (Direct I/O instructions) نقل مباشر.
- الوصلات والأسلاك التي تربط بين الأجهزة المختلفة.

ثالثا: تطبيقات المدخلات والمخرجات:

مثل تطبيق الانتقال من الأجهزة لنظام التشغيل.

تختلف الأجهزة في كثير من الأبعاد و التي تؤثر في التطبيق:

- (Character-stream or block) طابع التيار أو الكتلة
- (Sequential or random-access) التسلسل والترتيب أو الوصول العشوائي.
- (Synchronous or asynchronous) التزامن في التوقيت أو عدم التزامن.
- (Sharable or dedicated) التقاسم والتشارك أو التخصص في شئ والتكريس.
- (Speed of operation) السرعة في العمليات والتشغيل.

● أنه يكون للقراءة والكتابة أو قراءة فقط أو كتابة فقط. (read-write, read only, or write only)

رابعا: تحويل متطلبات المدخلات والمخرجات إلى عمليات على الأجهزة:

نظرة بشكل عام في قراءة ملف من القرص التي تتم عليه العملية....

1. تحديد الجهاز الذي سيحمل الملف.
2. ترجمة الاسم للجهاز الممثل .
3. قراءة البيانات من القرص العازل (buffer) إلى (disk) فيزيائيا.
4. جعل البيانات متاحة لمتطلبات العملية.
5. إعادة المراقبة للعملية.

خامسا: الأداء:

لتحسين الأداء:

1. خفض عدد المفاتيح (switches).

2. الحد من نسخ البيانات.

3. خفض الانقطاع (interrupts) باستخدام تحويلات كبيرة.

4. توازن وحدة المعالجة المركزية (CPU), الذاكرة, الناقل.

كتبته:

ناديا العتيبي

المصدر

- Operating System Concepts by Silberschatz, Galvin and Gagne

الإدخال والإخراج في نظام اللينكس

يوجد طريقتين في نظام اللينكس لمعالجة المدخلات والمخرجات داخل النظام

بالطريقة الأولى نستخدم نظام المناداة:

الخ open(), read(), write() مثل:

بينما بالطريقة الأخرى:

ANSI C تستخدم مكتبة مناداة ال

مثل: fopen(), fread(), fwrite() الخ

إن مكتبة السي في الحقيقة تحيط حول نظام المناداة و إنما تدعم قدر من الفوائد :

أولاً: إنها تخفف الإخراج اتوماتيكيا , لذلك تقلل احتياجات المناداة لنظام المناداة الذي يحسن الأداء.
ثانياً: الوظائف الملائمة متوفرة مع مساعده بطريقة تعديل المخرجات قبل عرضها .

قسمت المخرجات والمدخلات إلى نوعين :

1.(System call I/O) نظام الإدخال والإخراج.

2.(stream I/O) طريقة جيدة جدا فهي تحسن أداء التطبيقات .

شروط الأخطاء:

على المرء إن يركز على اكتشاف الأخطاء ومعالجتها أثناء التعامل مع المدخلات والمخرجات فبرنامجك قد يعمل (بدون أي أخطاء) لكن عندما يحدث خطأ ما فإنه سوف يفسد برنامجك و لمنع حدوث الخطأ وخسارة البيانات من أخطاء المدخلات والمخرجات , وهي شروط الأخطاء هناك طرق كثيرة منها:

للتأكد من أن الملف المراد تم فتحه بالفعل . `Open()`1. تحتاج إلى فحص القيمة المرجعة من مناداة الوظيفة

للتأكد أن القرص الذي يُكتب فيه , لم يمتلئ أثناء كتابتك للبيانات.`write()`2.فحص القيمة المرجعة من الوظيفة

ولكن عملية الفحص بعد كل نظام مناداة عملية مملة جدا...و لجعل عملية فحص الأخطاء ببرنامجك أوتوماتيكية تستطيع الذهاب إلى وظائف تفحص برنامجك وتكتب رسالة تحريك بحدوث الخطأ إذا اكتشفه.

عدم عرقلة المدخلات والمخرجات:

دائماً في نظام المناداة عندما ننادي وظيفة معينة , تنتظر إلى أن يتنفذ الآكشن المطلوب عمله هذه الطريقة في نظام الإدخال والإخراج تسمى بعدم عرقلة المدخلات والمخرجات ومن المهم أن نتذكر أن في هذه الطريقة، يُعوذُ نداء الوظيفة فوراً بصرف النظر عن إكمال العمل المطلوب.

الذاكرة - خرائط المدخلات والمخرجات:

يعتبر معلماً هاماً من معالم لينكس فمع هذه الميزة سيكون الملف مرتب بشكل حرفي في منطقته بالذاكرة ، فعندما تدخل تلك الذاكرة ، فإن العملية المخصصة تؤدي تلقائياً إلى الملف المحدد وهكذا. فباستخدام هذه الخرائط تزيد السرعة بشكل ملحوظ. لذا ، فإن هذا الأسلوب من المدخلات والمخرجات هو المفضل لدينا عندما تقرأ بيانات بالجملة.

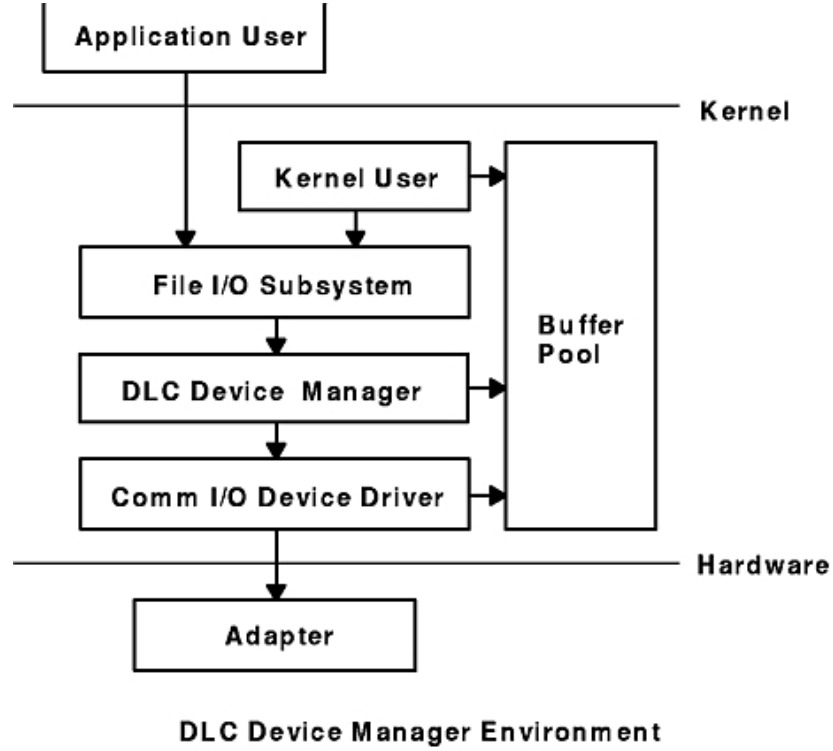
كتبته:

شيخة الرشود

المصدر:

[http://www.enterprisenetworksandservers.com/monthly/art.php?1503:](http://www.enterprisenetworksandservers.com/monthly/art.php?1503)

إدارة وحدات الإدخال والإخراج



وحدات الإدخال والإخراج تتضمن الوحدات الحقيقية مثل آلة الطباعة، الأشرطة المغناطيسية والأقراص المغناطيسية، وكذلك الوحدات مثل وحدة التحكم، القنوات، وحدات تحويل.

وتشرف على هذه الوحدات برمجيات خاصة تسمى برمجيات إدارة وحدات الإخراج والإدخال (Device management).

حيث تأخذ هذه البرمجيات على عاتقها تنفيذ المهام والوظائف التالية :

1. متابعة وحدات الإدخال والإخراج وذلك من خلال استخدام كتلة وحدة التحكم لكل وحدة

(control block unit(CBU)) التي تتضمن كافة المعلومات المطلوبة عن كل وحدة إدخال

وإخراج متصلة بالكمبيوتر.

2. اتخاذ سياسة معينة لحجز وحدات الإدخال أو الإخراج وفي زمن معين ولفترة زمنية محددة وغالبا ما تعتمد هذه

السياسة على طريقة استخدام وحدة الإدخال والإخراج (كوحدة غير مشتركة, مشتركة, إضافية)

3. الحجز الفيزيائي لوحدة الإدخال أو الإخراج وربط هذه الوحدة أو تلك مع العمل القابل للتنفيذ .

4. إعادة حجز وحدات الإدخال والإخراج المرتبطة مع العمل, وذلك بعد انتهاء تنفيذ العمل أو بعد

انتهاء الفترة الزمنية المخصصة للحجز.

طرق استخدام وحدات الإدخال والإخراج:

تقسم وحدات الإدخال والإخراج حسب طريقة الاستعمال إلى:

1. الوحدات غير المشتركة (Dedicated I/O Devices)

2. الوحدات المشتركة (shared I/O Devices)

3. الوحدات الافتراضية (Virtual I/O Devices)

الوحدات غير المشتركة:

وهي وحدات يمكن حجزها لعمل واحد فقط. ولا يمكن اشتراك أكثر من عمل قابل للتنفيذ في

استخدام هذه الوحدات, من الأمثلة على هذه الوحدات:

الآلات الطابعة, قارئ البطاقات المثقبة , وعند استخدام أي من هذه الوحدات من قبل عمل معين

فان الأعمال الأخرى القابلة للتنفيذ والتي قد تحتاجها الوحدة المحوزة عليها الانتظار حتى تتحرر

الوحدة المطلوبة عند انتهاء تنفيذ العمل أو عند انتهاء الفترة الزمنية المخصصة للعمل.

الوحدات المشتركة:

وهي وحدات يمكن استخدامها من قبل أكثر من عمل قابل للتنفيذ, وفي نفس الفترة الزمنية, ومن

الأمثلة على هذه الوحدات : وحدات الإدخال والإخراج ذات الوصول المباشر

(Direct Access Storage Device(DASD)).

وتعتبر إدارة . وتعتبر إدارة مثل هذه الوحدات أصعب من إدارة الوحدات غير المشتركة نظرا للحاجة إلى تحديد أولوية عملية القراءة أو الكتابة للأعمال المختلفة القابلة للتنفيذ كالقرص المغناطيسي مثلا.

الوحدات الافتراضية:

وهي وحدات غير مشتركة تم تحويلها إلى وحدات مشتركة, وذلك باستخدام تقنيات وبرمجيات خاصة كنظام التمرير (spooling system) حيث يؤدي هذا النظام إلى تحويل الوحدة غير المشتركة إلى وحدة مشتركة, فمثلا يمكن استخدام القرص المغناطيسي كآلة طباعة, مما يؤدي بدوره إلى تحقيق الفوائد التالية:

1. حل مشكلة عدد الوحدات غير المشتركة فمثلا لو كان لدينا عدة أعمال تحت التنفيذ وكلها تحتاج إلى عمليات طباعة, فإنه لتنفيذ هذه الأعمال لابد من توفر عدد من الطابعات مساويا لعدد الأعمال المنفذة مما يؤدي بدوره إلى تكاليف باهظة, وباستخدام نظام التمرير يمكن الاستعانة بالقرص المغناطيسي حيث تتم عملية الطباعة على القرص, وعند الحاجة تنقل المعلومات من القرص المغناطيسي إلى الطابعة.
2. زيادة سرعة المعالجة نظرا لأن سرعة تعامل وحدة المعالجة المركزية مع القرص المغناطيسي أعلى منها عند تعامل الطابعة مع وحدة المعالجة المركزية.

كتبته:

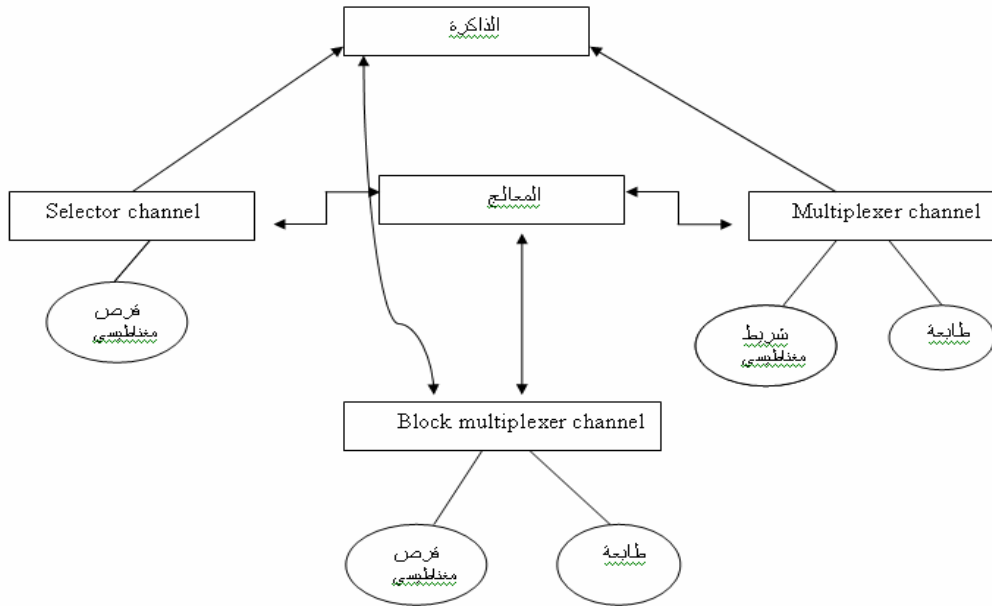
رزان المزروع

المصدر:

كتاب أنظمة التشغيل – للدكتور زياد القاضي.

قنوات الإدخال و الإخراج (IO\channels)

قنوات الإدخال و الإخراج عبارة عن أجهزة متخصصة يمكن أن تكون حاسوبا ، لكنها تخصص في الإشراف على ربط وحدات الإدخال و الإخراج مع المعالج أو مع الذاكرة لإتمام عمليات الإدخال والإخراج .
وتحتوي هذه الأجهزة (القنوات) على برنامج خاص يسمى برنامج القناة (program channels)
ووظيفته متابعة عمليات الإدخال و الإخراج والاتصال بين وحدة الإدخال / الإخراج والمعالج أو الذاكرة.
وتوضح الرسمة كيفية استخدام قنوات الإدخال و الإخراج لإجراء عمليات الربط اللازمة .



المصدر:

كتاب أنظمة التشغيل للدكتور زياد القاضي

المقاطعة (interrupt)

تعريفه : هو عبارة عند حدث في الوحدات الصلبة للجهاز, المساعدة في عمليات معالجة البيانات محفز للانتقال من مكان في البرنامج الحالي في العملية إلى مكان معين آخر.

طريقة الinterrupt الأساسية:

تحتوي الوحدة المعالجة المركزية على سلك يدعى **interrupt-request line** بحيث تتحسس بعد الانتهاء من كل أمر. فحين يكتشف المتحكم بالوحدة المعالجة المركزية إشارة على السلك, تخزن وحدة المعالجة المركزية الحالة الحالية وتنتقل إلى "مدير المقاطعة" **interrupt handler** وظيفة في عنوان محدد بالذاكرة. ويحدد **interrupt handler** سبب حدوث هذه المقاطعة (أي ظهور الإشارة) , وتقوم بعمل العمليات الضرورية وتخزين الحالة والانتهاء من العملية والعودة إلى العملية السابقة التي كانت قبل ظهور الإشارة. وبالتالي فعليا يقال الجهاز المتحكم **raises** "تحفز" المقاطعة عن طريق إحداث إشارة في السلك. ووحدة المعالجة المركزية **catches** "تقبض" المقاطعة وبالتالي ترسل "**dispatch**" إلى **interrupt handler** وهو بدوره يزيل **clear** المقاطعة عن طريق أداء الخدمة للجهاز .

في أنظمة التشغيل الحديثة نحتاج إلى مواصفات إضافية لل**interrupt handler** منها :

- 1_ نحتاج إلى القدرة على تأجيل **interrupt handler** خلال العمليات المهمة .
- 2_ نحتاج إلى طريقة ذات كفاءة عالية في إرسال ال **interrupt handler** الملائم للجهاز من غير عمل **polling** "التصويت" لجميع الأجهزة لمعرفة أيها منهم قام بتحفيز المقاطعة.
- 3_ نحتاج إلى **interrupt** ذو عدة مراحل حيث نظام التشغيل يستطيع التمييز من له الأولوية الأعلى عن المنخفضة حتى يستجيب للمقاطعة بالدرجة المناسبة له.

*في الأجهزة الحديثة قاموا بتزويدها بثلاث المواصفات الذي ذكرت سابقا, المزودة أصلا عن طريق الوحدة المعالجة المركزية وعن طريق الوحدة الصلبة المساعدة في معالجة البيانات للمتحكم بالمقاطعة "توقيف".

حيث يوجد microprocessors في الأجهزة الحديثة تقوم بمعالجة عملية المقاطعة .
فحينما نحدث عملية مقاطعة في احد أجهزة الجهاز فإن وحدة المعالجة المركزية تتوقف عن التنفيذ ثم تقفز إلى موقع الذاكرة التي تحوي رمز المقاطعة أو أمر رمز المقاطعة .عادة ما يشتغل هذا الرمز في نمط خاص من وحدة المعالجة المركزية .

تصنف بعض CPU المقاطعات في درجات من الأولوية حسب حدوثها .

كذلك تخصص بعض CPU سجلات خاصة للمقاطعات .A special set of registers
وبعدما تتم عملية المقاطعات فإن يتم استعادة لحالة CPU السابقة بحيث يكمل CPU عملياته التي توقفت عنها بسبب المقاطعة

كتبته

ماجدة بن طالب

المصدر:

<http://www.science.unitn.it/~fiorella/guidelinux/tlk/node81.html>

:

المقاطعة (interrupt)

الـ interrupt (مقاطعه): هو حدث في أجهزة الـ hardware يجعل المعالج يقفز في برنامجه الحالي إلى نقطة محددة في هذا code.

*الـ interrupt (مقاطعه): صممت لتكون أحداث خاصة لا يمكن التنبؤ بها بدقة (أو على الإطلاق).

* MSP تحتوي على العديد من الأنواع المختلفة من الأحداث التي يمكن أن تؤدي إلى interrupts ، وكل واحد من المعالجات سيرسل تنفيذ فريد إلى نقطة محددة في الذاكرة

يمكن تقسيم الـ interrupt (مقاطعه) بصفة عامة إلى نوعين :

1- maskable

2 non-maskable

Maskable interrupt : هي المقاطعة التي تؤدي إلى توقف عجلة الأحداث ولكنه ليس مهم دائما ، لذلك المبرمج يمكن أن يقرر إن هذا الحدث يجب ألا يسبب قفز إلى البرنامج.

A non-maskable interrupt : مهم جدا و لا ينبغي أبدا أن نتجاهله, المعالج دائما سيقفز إلى هذا interrupt (مقاطعه) عند حدوثه.

* في كثير من الأحيان ، maskable interrupt يتم إيقافه افتراضيا لتبسيط السلوك الافتراضي للجهاز.

*الـ interrupt (مقاطعه) عموما لها "الأولوية " عندما يحدث مقاطعتين (2 interrupts) في الوقت نفسه ، فإن الـ interrupt (مقاطعه) التي تملك أولوية أعلى سوف تكون لها الأسبقية على interrupt (مقاطعه) التي لها أولوية أدنى.

كتبته

غادة العندس

المصدر:

<http://cnx.org/content/m12321/latest/>:

أغلب وحدات المعالجة المركزية تحتوي على سلكين طلب لل interrupt :

1- الأول يدعى nonmaskable interrupt:

وهو محجوز للحدث مثل أخطاء الذاكرة الغير مصححة.

2- الثاني يدعى maskable interrupt:

وهو يمكنه الإيقاف عن طريق الوحدة المعالجة المركزية قبل الانتهاء من الأوامر المهمة التي من المفروض عدم إيقافها. وهو يستخدم عن طريق المتحكم بالجهاز لطلب الخدمات.

كتبته

زاهية الحربي

المصدر

Operating System Concepts by Silberschatz, Galvin and Gagne 7th edition

المقاطعة (interrupt)

Interrupt request line:

هو سلك بين CPU & controller يقوم CPU بفحصه بعد تنفيذ كل أمر لمعرفة ما إذا controller قد وضع فيه شحنة ما

عندما يجد CPU إن هنالك شحنة على السلك فعندها يعمل ما يعرف ب context switch فيعمل حفظ لما كان يعمل ثم يذهب إلى Interrupt handler الموجود في مكان ثابت في memory

Interrupt handler يقوم بتحديد حالة Interrupt و يقوم بعدة عمليات ضرورية لمعالجتهم ثم يرجع إلى CPU ليكمل ال CPU ما كان يفعله وتستمر الدورة .

يوجد لدينا نوعان من أسلاك interrupt:

1- Non-maskable interrupt:

هذا النوع لأجل الأخطاء التي يجب معالجتها في الحال مثل عنوان في الذاكرة لا يسمح بالدخول له.

2. Maskable.

يستطيع CPU أن يقفل هذا الخط لكي لا يتم إزعاجه خلال قيامه بتنفيذ الأوامر وهو يستخدم عادة لخدمة أجهزة الإخراج والإدخال.

Interrupt vector:

يحتوي على عناوين في الذاكرة لمجموعة مميزة من Interrupt handler وذلك لتقليل من عملية البحث ما إذا كان هنالك يوجد فقط handler واحد يقوم بالبحث عن جميع مصادر interrupt لمعرفة من يريد الخدمة.

Interrupt priority levels:

هذه التقنية تمكن CPU من التفريق بين المقاطعات ذات الأهمية العليا والتي ذات الأهمية الدنيا

نظام التشغيل يتعامل مع تقنية المقاطعات بعدة طرق:

1. Boot time فان CPU يحتاج لمعرفة من المتصل به من أجهزة الإخراج والإدخال وذلك لتعبئة

Interrupt vector

2. Exception وهي الأخطاء التي تأتي من Software مثل القسمة على صفر .

كتبته

نوف الغانمي.

المصدر:

: Operating System Concept

المقاطعة (interrupt)

مفهوم الاعتراض واستخدامه في عمليات الإدخال والإخراج:

الاعتراض هو حدث استثنائي يضطر نظام لتشغيل عند الاستجابة له وقف تنفيذ العمل الخاضع للتنفيذ من أجل تنفيذ عمل جديد يسمى العمل المعترض

وتمتلك نظم التشغيل برمجيات خاصة لمعالجة الاعتراض تسمى بمعالج الاعتراض `interrupt handler` وتتولى هذه البرمجيات تنفيذ الوظائف التالية:

- 1.. إصدار الاستجابة الموجبة والسالبة بناء على طلب الاعتراض من وحدة الإدخال والإخراج.
2. وفي حالة تقبل طلب الاعتراض عليه حفظ حالة البرنامج الخاضع للتنفيذ ولنسمة العمل القديم وذلك للرجوع إليه لاحقاً
3. يتم تحميل حالة العمل المعترض الجديد بعد وقف تنفيذ العمل القديم.
4. ينفذ العمل الجديد
5. بعد الانتهاء من العمل الجديد تحمل حالة البرنامج القديم لمتابعة تنفيذه من النقطة التي قطع منها.

اعتراض الإدخال والإخراج `IO interrupt`:

وينشأ هذا الاعتراض في قناة الإدخال أو الإخراج أو وحدات الإدخال والإخراج وأسبابه واحد مما يلي أو أكثر:

1. خطأ في أمر الإدخال أو الإخراج كأن يكون الأمر غير صحيح لغوياً أو غير متبع لقواعد كتابة أوامر الإدخال أو الإخراج أو خطأ في شفرته الثنائية أو عدم تحديد هذا الأمر كلا من رقمي وحدة الإدخال والإخراج ورقم القناة.

2. انتهاء تنفيذ برنامج الإدخال أو الإخراج وذلك بعد تنفيذ كلمة **command word** **channel** في برنامج القناة حيث يصدر نظم التشغيل الأمر المهيمن **HIO** لإيقاف برنامج القناة وبذلك يحدث اعتراض مفاجئ على القناة لتتوقف وتعمل أخرى في الحاسوب.
3. انتهاء وحدتي الإدخال أو الإخراج من عملية تمرير ونقل البيانات وفي هذه الحالة يتحكم برنامج ضابط التحكم بالإدخال والإخراج من ضبط عملية نقل البيانات بالاتجاهات المختلفة وذلك بالتعاون مع نظام التمرير **Spooling system** فإذا تمت عملية تمرير البيانات حدث اعتراض مفاجئ.

المصدر:

أنظمة التشغيل للدكتور زياد القاضي

الوصول المباشر للذاكرة (Direct Access Memory)

Direct Memory Access: هو نظام يسمح بنقل البيانات مباشرة من الذاكرة إلى أنظمة الإدخال والإخراج دون الحاجة لإذن من ال **CPU**.

Module: تنقل البيانات من موقع ذاكرة إلى موقع ذاكرة آخر.

* حالات الوصول إلى الذاكرة بشكل آلي أسرع بكثير من أن تدير **CPU** الانتقالات.

* الأنظمة **ADC, DAC, PWM**, تأسر كل متطلبات تحركات ال **memory** المتكررة والمنتظمة خارج أنظمتهم الخاصة بهم ..

* **DMA** : يمكن استخدامها لمعالجة انتقال البيانات المجمعة خارج الوحدات الخارجية إلى مواقع ذاكرة مفيدة أكثر.

Memory هي الوحيدة التي يمكنها أن تدخل هذا الطريق، لكن أكثر الأنظمة الخارجية peripheral systems وسجلات البيانات data registers والسجلات المتحكممة control registers يمكنها أن تدخل هذا الطريق كأنه ذاكرة..
DMA تستخدم عندما تكون الكهرواء منخفضة لأنها تستخدم نفس memory bus كوحدة المعالجة المركزية و فقط واحد أو الآخرون يمكنهم استخدام memory في نفس الوقت.

DMA: منظم إلى ثلاثة أجزاء مستقلة.. ومع ذلك فإن الثلاثة تتنافس على نفس memory bus ويمكن أن يشكلوا ال independent triggers , memory regions

هناك ثلاث قنوات مستقلة لانتقالات DMA .. كل قناة تستقبل trigger حتى ترسلها لأكبر عدد من ال signals المختارة.. عندما تكون هذه ال signals نشطة يتم الإرسال ..

DMA controller : يستقبل trigger signal ولكنه يهملها تحت شروط معينة.. وهذا ضروري لحجز memory bus لإعادة البرمجة و non-maskable interrupts و Controller يعالج التعارضات لل simultaneous triggers

كتبته

رهام حافظ

المصدر:

[:/http://cnx.org/content/m11867/latest](http://cnx.org/content/m11867/latest)

مشكلة ال(DMA) مع الذاكرة الوسيطة:

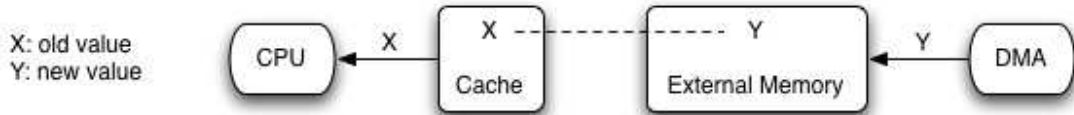
إذا كانت وحدة المعالجة المركزية مجهزة بذاكرة وسيطة وذاكرة خارجية التي يمكن أن تُدخل مباشرة بواسطة الأجهزة التي تستخدم (DMA) .

عندما تدخل وحدة المعالجة المركزية الموقع (س) في الذاكرة فإن القيمة الحالية لـ(س) ستخزن في الذاكرة الوسيطة.

والعمليات اللاحقة على(س) ستحدث النسخة الموجودة في الذاكرة الوسيطة بينما النسخة الموجودة في الذاكرة الخارجية لم تُحدث بعد.

لو حدث وقرأ أحد الأجهزة الموقع (س) فقبل أن تحدث نسخة الذاكرة الخارجية فإنه سيقراً فإن القيمة الغير محدثة.

وبنفس الطريقة لو كتب الجهاز في الذاكرة الخارجية ولم تُحدث نسخة الذاكرة الوسيطة فإن وحدة المعالجة ستقرأ قيمة غير مُحدثة



كتبته

صفاء البسام

http://en.wikipedia.org/wiki/Direct_memory_access: المصدر

ذاكرة الوصول المباشر (DMA):

يقدم CPU الكثير من الأعمال , فهو يشغل Bios ونظام التشغيل والتطبيقات . كما يعالج المقاطعات وعناوين I/O وأيضاً يعالج CPU كل البيانات بشكل دائم إذ يقوم بنقلها من RAM إلى مكان آخر كما ترسل الطرفيات , مثل الطابعة والمسح الضوئي البيانات إلى RAM عبر CPU , وترسل CPU البيانات من RAM إلى الطرفيات .

ومن الواضح أهمية نقل هذه البيانات , لكنها عملية بسيطة في النهاية , ولدى CPU أعمال أكثر أهمية لتقوم به.

علاوة على ذلك , ومع كل هذه الذاكرة المخبئة Cache memory في CPU الحديثة يهدر النظام معظم وقته في انتظار معالجة بعض الحسابات الداخلية في CPU.

ولذلك يطرح السؤال التالي : لماذا لا يوجد جهاز يقوم بالوصول المباشر إلى الذاكرة دون تدخل CPU?

تدعى عملية الوصول المباشر إلى الذاكرة بدون تدخل CPU : Direct Memory Access :

وهي سمة من سمات أجهزة الكمبيوتر الحديثة التي تسمح لبعض الأنظمة الفرعية داخل أجهزة الكمبيوتر للوصول إلى نظام لذاكره القراءة و / أو الكتابة بصورة مستقلة من وحدة المعالجة المركزية ((CPU)).

تتيح DMA تشغيل التطبيقات في الخلفية , بدون تدخل CPU . وهذا طبعاً ممتاز لتشغيل الصوت في الخلفية , ونقل البيانات من القرص المرن أو القرص الصلب إلى RAM.

العديد من أنظمة hardware تستخدم DMA والتي تشمل متحكمي تشغيل القرص ((disk drive controllers)) وبطاقات الرسوم وبطاقات الشبكة وبطاقات الصوت ..

أجهزة الكمبيوتر التي لديها DMA channels تستطيع نقل البيانات من وإلى الأجهزة بتكلفة أقل بكثير من وحدة المعالجة المركزية ((CPU)) ..

مع **DMA** , وحدة المعالجة المركزية ((CPU)) ستبدأ بالنقل , وإجراء العمليات الأخرى بينما النقل مستمر , وكذلك استقبال القاطع ((interrupt)) من متحكم الـ **DMA** عندما تكتمل العملية ,

هذا جدا مفيد وخصوصا في التطبيقات الحاسوبية للوقت الحقيقي ((real-time)) وبذلك لا يكون هناك تعطيل للعمليات المتزامنة ..

المشكلة هنا : ما العمل إذا طلب أكثر من جهاز استخدام **DMA** ؟ كيف نمنع تزامن الأجهزة على ممر البيانات الخارجي ؟ ماذا لو احتاجت **CPU** إلى ممر البيانات فجأة ؟ كيف يمكن إيقاف جهاز يستخدم **DMA** بحيث تستطيع **CPU** (ذات الأولوية) الوصول إلى الممر ؟

ولمعالجة ذلك قامت **IBM** بإنشاء رقاقة تدعى رقاقة 8237 للتحكم بعمليات **DMA**. تستطيع هذه الرقاقة معالجة كل عمليات نقل البيانات من الطرفيات إلى **RAM** , أو بالعكس . وهذا ما يوفر من وقت وجهد الـ **CPU** .

كتبته

مي الغيث - هند المطيري

المصدر

[http://en.wikipedia.org/wiki/Direct_memory_access:](http://en.wikipedia.org/wiki/Direct_memory_access)

الاستطلاع (polling)

ماهية فكرة الاستطلاع (Polling)؟

لها الكثير من المصطلحات ولكن نستطيع تشبيهها بالايمل . فعندما يصل الايمل للشخص فانه يتجمع في الصندوق الوارد . ويقوم بعدها الشخص (صاحب الايمل) بفحص أيميله كل فترة لمعرفة الجديد في صندوقه الوارد .

هذه فكرة مبسطة عن الاستطلاع (poll) فـجهاز الكمبيوتر يحتوي على العديد من أجهزه الإدخال والإخراج المتصلة به وكلها تحتاج في وقت ما إلى محادثة المعالج .

عن طريق هذه الفكرة فان المعالج يقوم كل فترة زمنية بفحص جميع أجهزة الإخراج والإدخال المتصلة به لمعرفة ما إذا هنالك جهاز يحتاج إلي خدمة ما فيقوم بخدمته

Polling

هو بروتوكول للتفاعل بين المضيف (CPU) و المتحكم (controller).

Controller يصف حالته عن طريق Busy Bit الموجود في status register

controller عندما يكون مشغول يعمل write 1 on it) set busy bit) وعندما يكون جاهز

لاستقبال الأوامر فانه يعمل (write 0 on it) clear to busy bit

هذا بالنسبة لل controller كيف يعبر عن حالته أما بالنسبة إلى CPU فانه يصف حالته

إلى controller عن طريق command-ready bit in command register

التعامل بينهم يكون كالتالي :

المضيف يفحص باستمرار Busy bit حتى يجده ب 0 فيعلم أن controller جاهز لاستقبال الأوامر المضيف يعمل set to write bit in command register ثم يكتب byte في data-out register يعمل المضيف set command-ready bit

عندما يلاحظ controller بان ready bit has been set يقوم بعمل set to busy bit

ثم يقول controller بقراءة البيانات الموجودة في command register ويجد أن write قد عمل له set (قيمته ب 1)

فيعلم بان عليه أن يرسل هذه البيانات إلى جهاز إخراج

ثم يقوم ال controller بعمل مسح (clear) لل command ready bit وأيضا يقوم بمسح error bit يعلم المضيف بان عملية الكتابة تمت بنجاح وأيضا يقوم بمسح busy bit ليعلمه بان مهمته انتهت ثم تقوم هذه الدورة بالاستمرار.

كتبته

نوف الغانمي

المصدر

: Operating System Concept