

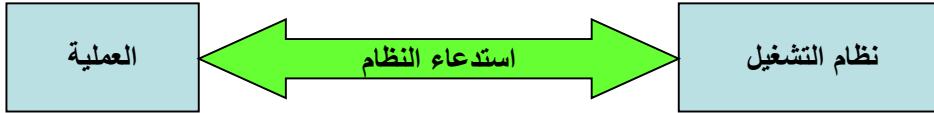
الاسبوع الرابع عشر

نداءات (إستدعاء) النظام

System Calls

نداءات (إستدعاء) النظام System Calls

برنامج استدعاء النظام يوفر اتصال بين العملية ونظام التشغيل ، وهي عبارة عن برامج مكتوبة باللغة التجميعية (Assembly Language) . وكما موضح في الشكل (23) .



الشكل (23) يوضح علاقة العملية مع نظام التشغيل من خلال الاستدعاء

يمكن تقسيم إستدعاءات النظام إلى خمسة أنواع:

- 1- إنهاء العملية بشكل طبيعي .
- 2- إنهاء أو إيقاف غير طبيعي للعملية .
- 3- إستدعاء طلب الحالة : قد يكون هذا النوع من الاستدعاء للسؤال عن معلومات أو حالة النظام ، كأن يكون إستدعاء لمعرفة معلومات عن التاريخ والزمن ونوع الملفات وعدد المستخدمين في الحاسوب ومقدار الذاكرة .
- 4- إستدعاء طلب الموارد : عند وجود حاجة إلى مورد من موارد الحاسوب من قبل عملية معينة لغرض الاستمرار في التنفيذ لذا يحصل إستدعاء للمورد لغرض الحجز والتنفيذ .
- فمثلا عند وجود حاجة لحجز جهاز الطابعة من قبل عملية ، فإذا كانت الطابعة متوفرة تستمر العملية بشكل طبيعي في التنفيذ والطبع ، أما إذا كانت الطابعة غير متوفرة فإن المستخدم أو العملية تبقى بالانتظار لحين توفرها .
- 5- إستدعاء طلب الإدخال/الإخراج : يعتبر هذا النوع أكثر شيوعا واستخداما لأن معظم العمليات تحتاج إلى إدخال وإخراج بسبب القراءة والكتابة من وإلى الأجهزة والملفات .

مثال : نسخ ملف

- قراءة اسم الملف الأول ثم قراءة اسم الملف الثاني .
- فتح الملف الأول للقراءة وفتح الملف الثاني للكتابة (مع معالجة الأخطاء) .
- قراءة محتويات الملف الأول وكتابتها إلى الثاني .
- إغلاق الملفين .
- إنهاء البرنامج .

دعوات النظام الخاصة بإدارة العمليات : System Calls for Process management

- الاستدعاء Fork يقوم بإنشاء عملية جديدة في نظام UNIX ، يسبب تنفيذ Fork إنشاء نسخة مطابقة من العملية (الأب) تسمى (الابن) وتختلف عنها فقط في رقم العملية PID ، يقوم الاستدعاء Fork بعد التنفيذ بإرجاع قيمتين:
- في العملية (الأب) : يعيد الاستدعاء Fork رقم العملية (الابن) كنتيجة له.
 - في العملية (الابن) : يعيد الاستدعاء Fork قيمة (0) كنتيجة له.
- يتابع كلا من العملية (الأب) والعملية (الابن) التنفيذ ابتداءً من الأمر الذي يلي Fork

العملية (الأب)	العملية (الابن)
PID = x	PID = y
.....
.....
Child = fork(); ← y	child = fork(); ← 0
If child > 0	If child > 0
{...parent instructions...}	{...parent instructions...}
else	else
{...child instructions...}	{...child instructions...}

حتى تساعد على تمييز العمليتين المتماثلتين :

- العملية (الأب) إما :

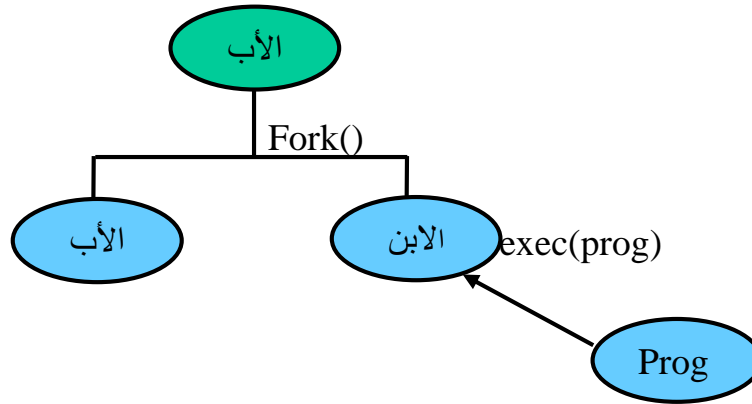
* تنتظر انتهاء تنفيذ العملية (الابن).

* العملية (الأب) تواصل تنفيذها.

- إذا قررت العملية (الأب) انتظار العملية (الابن) فإن الأب سيستقبل أمر الخروج من العملية التي من الابن. وفي الحالة الأخرى :

فإن الأب يعمل على تجاهل العملية (الابن) ويستمر الأب في تنفيذ عمله .

الاستدعاء Exec ، العملية (الابن) تقوم بمناداة exec والتي تعمل على تبديل البرنامج الذي يحتويه الابن بنسخة جديدة من برنامج آخر. الشكل (24) يوضح علاقة العملية الابن مع العملية الأب من خلال الاستدعاء Exec .



الشكل (24) علاقة العملية الابن مع العملية الأب من خلال الاستدعاء Exec

الجدول (5) أدناه يبين وصف لبعض الاستدعاءات الخاصة بإدارة العمليات في نظام UNIX

الاستدعاء	الوصف
Pid = fork()	إنشاء عملية ابن مطابقة للعملية الأب.
Pid = waitpid(pid , &statloc , options)	انتظار ابن حتى ينتهي.
S = exec(name , argv , environp)	استبدال صورة الذاكرة لعملية ما.
Exit(status)	إنهاء تنفيذ عملية وإعادة الحالة.

دعوات النظام الخاصة بإدارة الملفات : System Calls for file management

هناك العديد من استدعاءات النظام المتعلقة بنظام الملفات حصراً. لقراءة ملف أو الكتابة إليه ، يجب أن يفتح الملف أولاً باستخدام `open` . يحدد هذا الاستدعاء اسم الملف المراد فتحه، إما بمسار مطلق أو مسار نسبي للدليل الحالي، وتعني (`O_RDONLY` و `O_WRONLY` و `O_RDWR`) أن الفتح للقراءة فقط أو للكتابة فقط أو لكليهما. لإنشاء ملف جديد نستخدم الأمر `O_CREAT` ، يمكن إغلاق الملف بواسطة الاستدعاء `close`. تعتبر `read` و `write` استدعاءات النظام الأكثر استخداماً بلا شك. أما `write` فإن له نفس البارامترات.

مع أن معظم البرامج تقرأ الملفات وتكتبها بشكل تسلسلي ، فإن بعض البرامج التطبيقية تحتاج لإمكانية الوصول العشوائي إلى أي جزء من الملف. يرتبط بكل ملف مؤشر يحدد الموقع الحالي في الملف. عند قراءة (أو كتابة) الملف تسلسلياً. يشير هذا المؤشر إلى البايت التالي. يغير الاستدعاء `Iseek` قيمة مؤشر الموقع، وبالتالي يمكن لاستدعاءات `read` و `write` أن تبدأ من أي مكان في الملف.

يملك الاستدعاء `Iseek` ثلاثة بارامترات: الأول واصف الملف `fd` والثاني موقع الملف `offset` والثالث يدل إذا ما كان موقع الملف منسوباً إلى بداية الملف أم للموقع الحالي أم لنهاية الملف `whence` . يعيد الاستدعاء `Iseek` موقع الملف المطلق بعد تغيير المؤشر.

يحتفظ UNIX من أجل كل ملف بمعلومات عن نمطه (ملف عادي أم ملف خاص أم دليل) وحجمه وتاريخ التعديل الأخير وغيرها من المعلومات. تستطيع البرامج الاستعلام عن هذه المعلومات بواسطة استدعاء النظام `stat` . يحدد البارامتر الأول الملف المراد السؤال عنه والثاني عبارة عن مؤشر إلى كتلة البيانات التي ستوضع فيها المعلومات.

الجدول (6) وصف لبعض الاستدعاءات الخاصة بإدارة الملفات في نظام UNIX

الوصف	الاستدعاء
فتح ملف للقراءة أو الكتابة أو كليهما.	<code>fd = open(file , how ,)</code>
إغلاق ملف مفتوح.	<code>s = close(fd)</code>
قراءة البيانات من ملف إلى مخزن مؤقت.	<code>n = read(fd , buffer , nbytes)</code>
كتابة بيانات إلى ملف من مخزن مؤقت.	<code>n = write(fd , buffer , nbytes)</code>
تحريك مؤشر الملف.	<code>Position = lseek(fd , offset , whence)</code>
الحصول على معلومات حالة الملف.	<code>s = stat(name , &buf)</code>

دعوات النظام الخاصة بإدارة الأدلة والمجلدات: System Calls for Directory management:

سوف نتطرق إلى بعض استدعاءات النظام المتعلقة أكثر بالأدلة أو بنظام الملفات ككل عوضاً عن التعامل مع ملف معين . يقوم الاستدعاءات `mkdir` و `rmdir` بإنشاء وإزالة دليل فارغ على الترتيب . الاستدعاء التالي هو `link` ، ويسمح بظهور ملف واحد باسمين أو أكثر، ويكون ذلك عادةً في أدلة مختلفة. من استخداماته العملية السماح لعدة مستخدمين بالوصول إلى ملف مشترك بحيث يظهر هذا الملف في الدليل الخاص بكلٍ منهم، وربما بأسماء مختلفة. لأن التشارك على ملف يعني أن أي تغيير يجريه أحد المستخدمين سيراه باقي المستخدمين على الفور لأن هناك نسخة واحدة فقط من الملف.

الاستدعاء `mount` يسمح بدمج نظامي ملفات ضمن نظام ملفات واحد. عندما تنتهي الحاجة لنظام الملفات، يمكن إزالة تثبيته بواسطة استدعاء النظام `unmount`.

الجدول (7) وصف لبعض الاستدعاءات الخاصة بإدارة الأدلة ونظام الملفات في نظام UNIX

الوصف	الاستدعاء
إنشاء دليل جديد.	<code>s = mkdir(name , mode)</code>
إزالة دليل فارغ.	<code>s = rmdir(name)</code>
إنشاء مدخل جديد <code>name2</code> يشير إلى <code>name1</code> .	<code>s = link(name1 , name2)</code>
إزالة مدخل من دليل.	<code>s = unlink(name)</code>
تثبيت نظام ملفات.	<code>s = mount(special , name , flag)</code>
إزالة تثبيت نظام ملفات.	<code>s = unmount(special)</code>

اسئلة اختبارية:

- س1: عدد انواع استدعاءات النظام ؟
- س2: برنامج استدعاء النظام يوفر اتصال بين 000000 و 000000 ومكتوبة باللغة 0000000 .
- س3: ما هي وظيفة الاستدعاء Fork في ادارة العمليات في نظام التشغيل Unix ؟
- س4: لفتح ملف يتم استخدام الاستدعاء 000000 ويكون الفتح لـ 0000 فقط أو 00000 فقط أو 00000000 ؟
- س5: ماذا يعني الاستدعاء Iseek في ادارة نظام الملفات ؟
- س6: ماذا يعني الاستدعاء link في ادارة نظام الادلة والمجلدات ؟