

Chapter 3: Processes

نظام التشغيل ينفذ مجموعة متنوعة من البرامج:

- نظام الدفعات – الوظائف (Batch system)
- أنظمة المشاركة الزمنية – برامج المستخدم أو المهام (Time-shared system)

العملية Process

برنامج قيد التنفيذ؛ يجب أن تتقدم عملية التنفيذ بطريقة تسلسلية.

أجزاء العملية:

١. **text section** وهي التي تحتوي على الكود البرمجي وتسمى أيضًا قسم النص.
٢. **program counter** يحتوي على النشاط الحالي للعمليات ويخزن عناوين العمليات التالية وسجلات المعالج.
٣. **Stack** المكسدس وهو الذي يحتوي على البيانات المؤقتة:
٤. **قسم البيانات Data section** الذي يحتوي على المتغيرات العالمية.
٥. **الذاكرة الديناميكية (Heap)** التي تحتوي على الذاكرة المخصصة ديناميكيًا أثناء وقت التشغيل.

ملاحظة

- ١) يطلق على البرنامج الذي لم يتم تشغيله بعد حيث يكون مخزن في الـ Hard Disk بالـ **Passive entity**
- ٢) يطلق على البرنامج عند تشغيله وانتقاله لذاكرة RAM بالـ **Active entity**

لتشغيل برنامج هناك طريقتين:

١. من خلال النقر عليه بالماوس بالتفاعل مع واجهة المستخدم GUI
٢. من خلال موجه الأوامر CML

العملية الجزئية: Threads

هي وحدة التنفيذ داخل العملية. يمكن أن تحتوي العملية على عملية جزئية واحد فقط أو العديد من العمليات الجزئية.

حالات العملية:

- ١) جديد (new): يتم إنشاء العملية.
- ٢) قيد التشغيل (running): يتم تنفيذ التعليمات.
- ٣) انتظار (waiting): العملية تنتظر حدوث حدث معين.
- ٤) جاهز (ready): العملية تنتظر أن تُخصص لمعالج.
- ٥) منته (terminated): انتهت العملية من التنفيذ.

كتلة التحكم بالمهام (PCB) Process Control Block

هي عبارة عن جدول يحتوي على المعلومات المرتبطة بكل عملية. وتتضمن معلومات عن التالي:

- حالة العملية **Process state**: قيد التشغيل، انتظار، إلخ.
- عداد البرنامج **Program counter**: موقع التعليمات التالية التي سيتم تنفيذها.
- سجلات المعالج **CPU registers**: محتويات جميع السجلات المتعلقة بالعملية.
- معلومات جدولة المعالج **CPU scheduling information**: الأولويات، مؤشرات قائمة الجدولة.
- معلومات إدارة الذاكرة **Memory-management information**: الذاكرة المخصصة للعملية.
- معلومات الحسابات **Accounting information**: وحدة المعالجة المركزية المستخدمة، الوقت المنقضي منذ البداية، حدود الوقت.
- معلومات حالة الإدخال/الإخراج **I/O status information**: أجهزة الإدخال/الإخراج المخصصة للعملية، قائمة الملفات المفتوحة.

جدولة العمليات Process Scheduling

- **time sharing** هو التبديل السريع بين العمليات على وحدة المعالجة المركزية لتوفير زيادة استخدام وحدة المعالجة المركزية CPU.

- **جدول العمليات**: هو أن يقوم المعالج CPU بخيار من بين العمليات المتاحة للتنفيذ التالي على وحدة المعالجة المركزية.

- الحفاظ على قوائم جدولة العمليات:

- قائمة الوظائف **(Job queue)**: مجموعة من جميع العمليات في النظام.
- قائمة العمليات الجاهزة **(Ready queue)**: مجموعة من جميع العمليات الموجودة في الذاكرة الرئيسية، جاهزة وتنتظر التنفيذ تكون موجودة في **Program counter**.
- قوائم الأجهزة **(Device queues)**: مجموعة من العمليات التي تنتظر جهاز الإدخال/الإخراج.

تمثيل جدولة العمليات Representation of Process Scheduling

حيث يقوم مخطط الطوابير (Queueing diagram) بتمثيل الطوابير والموارد والتدفقات.

الجدولة (Schedulers)

١. المجدول قصير الأجل (Short-term scheduler) (أو مجدول وحدة المعالجة المركزية):

- يحدد أي عملية يجب أن تُنفذ بعد ذلك ويخصص وحدة المعالجة المركزية لها.
- يتم استدعاء المجدول قصير الأجل بشكل متكرر (كل بضع ميلي ثانية) مما يتطلب أن يكون سريعًا.

٢. المجدول طويل الأجل (Long-term scheduler) (أو مجدول الوظائف):

- يحدد أي العمليات يجب إدخالها إلى قائمة العمليات الجاهزة.
- يتم استدعاء المجدول طويل الأجل بشكل غير متكرر (كل ثوانٍ أو دقائق)، مما يسمح له بأن يكون بطيئًا.

تصنيف العمليات:

- عملية معتمدة على الإدخال/الإخراج (I/O-bound process):

- تقضي وقتاً أطول في إجراء عمليات الإدخال/الإخراج مقارنةً بالحسابات، مع العديد من فترات وحدة المعالجة المركزية القصيرة.

- عملية معتمدة على وحدة المعالجة المركزية (CPU-bound process):

- تقضي وقتاً أطول في إجراء الحسابات، مع فترات وحدة معالجة مركزية طويلة جداً وقليلة.

٣. جدولة متوسطة الأجل (Medium Term Scheduling)

(a) المجدول متوسط الأجل (Medium-term scheduler) يمكن إضافته إذا كانت هناك حاجة لتقليل درجة تعدد البرمجة.

(b) إزالة العملية من الذاكرة: وتخزينها على القرص. ثم إعادتها من القرص لاستئناف التنفيذ، وهو ما يعرف بـ **التبديل (swapping)**.

تبديل السياق (Context Switch)

هو عندما تقوم وحدة المعالجة المركزية بالتبديل من عملية إلى عملية أخرى، حيث يجب على النظام حفظ حالة العملية القديمة وتحميل الحالة المحفوظة للعملية الجديدة من خلال تبديل السياق.

ملاحظة:

- السياق الخاص بعملية ما: يتم تمثيله في كتلة التحكم في العملية (PCB).

- **زمن تبديل السياق Context-switch time**: يُعتبر عبئاً على النظام؛ حيث لا يقوم النظام بأي عمل مفيد أثناء عملية التبديل. كلما زادت تعقيدات نظام التشغيل وكتلة التحكم في العملية (PCB)، زاد زمن تبديل السياق.

الاجراءات على العمليات (Operations on Processes)

- إنشاء العمليات (Process creation)

- إنهاء العمليات (Process termination)

١. إنشاء العمليات (Process Creation)

- العملية الأم (Parent process) تنشئ عمليات فرعية (children processes)، التي بدورها يمكن أن تنشئ عمليات أخرى، مما يشكل شجرة من العمليات. حيث تُعرف العملية وتُدار عادةً عبر مُعرف العملية (Process Identifier - PID).

٢. خيارات مشاركة الموارد Resource sharing options:

يكون مشاركة الموارد بطريقتين بين العملية الأم والعمليات التابعة لها:

- (a) **مشاركة جميع الموارد**: العملية الأم والعمليات الجزئية التابعة children لها يشتركون في جميع الموارد.
- (b) **مشاركة مجموعة فرعية من الموارد**: العمليات الجزئية children يشتركون في مجموعة فرعية من موارد العملية الأم.

٣. خيارات التنفيذ Execution options:

- التنفيذ المتزامن: العملية الأم والعمليات التابعة لها children ينفذون بشكل متزامن.
- انتظار العملية الأم: قد تنتظر العملية الأم حتى تنتهي العمليات الفرعية children

٤. مساحة العنوان (Address space):

- تكون العملية الفرعية نسخة مكررة من العملية الأم.

أمثلة من نظام UNIX:

fork () : ينشئ عملية جديدة.

Exec() تنفيذ عملية في النظام.

٥. إنهاء العمليات (Process Termination)

- عندما تنتهي العملية من تنفيذها تقوم بطلب من نظام التشغيل حذفها باستخدام استدعاء النظام exit() .
- إرجاع حالة العملية: تُرجع بيانات الحالة من العملية الفرعية إلى العملية الأم الامر wait()
- إلغاء تخصيص الموارد: يتم إلغاء تخصيص موارد العملية من قبل نظام التشغيل عند اكتمال تنفيذ العملية.
- إنهاء العملية الأم العمليات الفرعية: يمكن للعملية الأم إنهاء تنفيذ العمليات الفرعية باستخدام استدعاء النظام abort() .

الحالات التي يجعل العملية الأم بإلغاء العمليات التابعة لها:

١. تجاوز العملية الفرعية الموارد المخصصة لها.
٢. المهمة الموكلة إلى العملية الفرعية لم تعد مطلوبة.
٣. إذا كانت العملية الأم تنهي تنفيذها، فإن نظام التشغيل لا يسمح للعملية الفرعية بالاستمرار إذا تم إنهاء والدها.

ملاحظة:

بعض أنظمة التشغيل لا تسمح بوجود عملية فرعية إذا كانت العملية الأم قد انتهت. أي إذا انتهت العملية الأم، فيجب إنهاء جميع عملياتها الفرعية أيضًا. وهذا ما يُعرف بـ **إنهاء متسلسل (cascading termination)**، حيث تُنهي جميع العمليات الفرعية، عند انتهاء عملية الأم.

ملاحظة:

يمكن للعملية الأم الانتظار لإنهاء عملية فرعية باستخدام استدعاء النظام wait() ويتم استرجاع معلومات الحالة للعملية المنتهية بواسطة PID

عملية الزومبي (Zombie process): هي عملية فرعية قد انتهت تنفيذها، ولكن لا تزال شغالة في النظام.

ويحدث هذا عند عدم قيام عملية الأم باستدعاء النظام wait() .

العملية اليتيمة (Orphan process): إذا انتهت العملية الأم دون استدعاء wait()، تصبح العملية الفرعية يتيمة.

التواصل بين العمليات (IPC) (Interprocess Communication)

العمليات في النظام: قد تكون العمليات مستقلة أو متعاونة.

- **العمليات المتعاونة (Cooperating processes):** هي عمليات يمكن أن تؤثر أو تتأثر بعمليات أخرى، بما في ذلك مشاركة البيانات.

- **العمليات المستقلة (Independent processes):** هي عمليات لا يمكن أن تؤثر أو تتأثر بتنفيذ عملية أخرى.

أسباب لوجود عمليات متعاونة:

- مشاركة المعلومات.
- زيادة سرعة الحساب.

- نموذجان للتواصل بين العمليات:

- الذاكرة المشتركة (Shared memory).
- تمرير الرسائل (Message passing).

١. المخزن غير المحدود (Unbounded buffer):

- لا يضع حدًا عمليًا على حجم المخزن.
- قد يضطر المستهلك إلى انتظار وجود عناصر جديدة، لكن المنتج يمكنه دائمًا إنتاج عناصر جديدة.

٢. المخزن المحدود (Bounded buffer):

- يفترض وجود حجم ثابت للمخزن.
- في هذه الحالة، يجب على المستهلك الانتظار إذا كان المخزن فارغًا، ويجب على المنتج الانتظار إذا كان المخزن ممتلئًا.

التواصل بين العمليات – الذاكرة المشتركة (Interprocess Communication – Shared Memory)

- منطقة الذاكرة المشتركة: هي منطقة من الذاكرة تُستخدم من قبل العمليات التي ترغب في التواصل.
- تحكم المستخدم: يتم التحكم في عملية التواصل من قبل العمليات المستخدمة، وليس من قبل نظام التشغيل.

التواصل بين العمليات – تمرير الرسائل (Interprocess Communication – Message Passing)

- هي آلية للتواصل تُستخدم لتواصل العمليات ومزامنة أفعالها. حيث يمكن أن تتواصل العمليات مع بعضها البعض دون الحاجة إلى استخدام متغيرات مشتركة.

- توفير خدمات IPC: يوفر نظام التواصل بين العمليات عمليتين رئيسيتين:

- send(message): لإرسال رسالة.

- receive(message): لاستقبال رسالة.

روابط التواصل بين العمليات: هي التي تحدد الجوانب كيفية تواصل العمليات مع بعضها البعض وكيفية إدارة الرسائل بشكل فعال.

- الروابط الفيزيائية:

- الذاكرة المشتركة (Shared memory)

- الموصلات المادية (Hardware bus)

- الشبكة (Network)

- الروابط المنطقية:

- مباشر أو غير مباشر (Direct or indirect)

- متزامن أو غير متزامن (Synchronous or asynchronous)

- تخزين تلقائي أو صريح (Automatic or explicit buffering)

التواصل المباشر (Direct Communication)

- حيث يجب على العمليات أن تسمى بعضها البعض بشكل صريح:

- send(P, message): إرسال رسالة إلى العملية P.

- receive(Q, message): استقبال رسالة من العملية Q.

خصائص رابط التواصل:

- يتم إنشاء الروابط تلقائيًا.

- يرتبط رابط واحد فقط بزوج محدد من العمليات المتواصلة.

- يوجد رابط واحد فقط بين كل زوج من العمليات.

- قد يكون الرابط أحادي الاتجاه، ولكن عادةً ما يكون ثنائي الاتجاه.

التواصل غير المباشر (Indirect Communication)

- حيث يتم توجيه الرسائل واستقبالها من صناديق البريد (mailboxes) (تُعرف أيضًا بالمنافذ - ports).
- يكون لكل صندوق بريد معرف فريد.
- يمكن للعمليات التواصل فقط إذا كانت تشارك صندوق بريد مشترك.

خصائص رابط التواصل:

- يتم إنشاء الرابط فقط إذا كانت العمليات تشارك صندوق بريد مشترك.
- قد يكون الرابط مرتبطًا بالعديد من العمليات.
- يمكن لكل زوج من العمليات أن يشارك عدة روابط تواصل.
- قد يكون الرابط أحادي الاتجاه أو ثنائي الاتجاه.

كيف تتم العمليات في الاتصال غير مباشر:

- إنشاء صندوق بريد جديد (port).
- إرسال واستقبال الرسائل عبر صندوق البريد.
- تدمير صندوق البريد.

عيوب التواصل غير المباشر

عدم معرفة من يتلقى الرسالة؟

الحلول:

1. السماح بارتباط الرابط بعملياتين بحد أقصى.
2. السماح لعملية واحدة فقط في كل مرة بتنفيذ عملية الاستقبال.
3. السماح للنظام باختيار المستلم بشكل عشوائي، مع إبلاغ المرسل بمن كان المستلم.

التزامن (Synchronization)

- تمرير الرسائل: يمكن أن يكون إما متزامنًا بشكل متكتل (block) أو غير متزامن (non-block).

- (Blocking): يُعتبر متزامنًا.

- إرسال متزامن: المرسل يكون محجوزًا حتى يتم استلام الرسالة.
- استقبال متزامن: المستلم يكون محجوزًا حتى تصبح رسالة متاحة.

- (Non-blocking): يُعتبر غير متزامن.

- إرسال غير متزامن: المرسل يرسل الرسالة ويستمر في التنفيذ.

- استقبال غير متزامن: المستلم يستقبل رسالة صالحة، أو رسالة فارغة (Null message).

حالة التقاء (rendezvous).

هي عندما يكون كل من الإرسال والاستقبال محجوزين، فإننا نكون في

التخزين المؤقت (Buffering)

يتم تنفيذ التخزين المؤقت بثلاث طرق:

١. سعة صفرية (Zero capacity):

- لا يتم تخزين أي رسائل على الرابط.

- يجب على المرسل الانتظار حتى يستقبل المستلم الرسالة (التقاء - rendezvous).

٢. سعة محدودة (Bounded capacity):

- طول محدود من n رسائل.

- يجب على المرسل الانتظار إذا كان الرابط ممتلئًا.

٣. سعة غير محدودة (Unbounded capacity):

- طول غير محدود.

- لا ينتظر المرسل أبدًا.

الاتصالات في أنظمة العميل-الخادم (Communications in Client-Server Systems)

المقابس (Sockets)

هو نقطة نهاية للتواصل وهو عملية دمج الـ IP address مع الـ Port Number.

ثلاثة أنواع من المقابس (Sockets)

١. مقبس معتمد على الاتصال (Connection-oriented - TCP):

- يوفر اتصالاً موثوقاً بين العميل والخادم.

- يضمن تسليم البيانات بترتيبها الصحيح.

٢. مقبس غير معتمد على الاتصال (Connectionless - UDP):

- يوفر طريقة سريعة لإرسال البيانات دون ضمان التسليم.

- لا يضمن ترتيب البيانات.

٣. مقبس متعدد MulticastSocket:

- تُستخدم لإرسال البيانات إلى مستلمين متعددين في وقت واحد.

- تدعم الاتصال الجماعي (multicasting) في الشبكات.

استدعاءات الإجراءات عن بُعد (RPC)(Remote Procedure Calls)

هو طريقة تستخدم لعملية التواصل بين الخادم server والعميل client.

يستخدم RPC المنافذ لتمييز الخدمات.

:(Stubs)

هو وكيل على جانب العميل لعمل إجراءات فعلية على الخادم.

وظائفه:

- **client-stubs** يقوم بتحديد موقع الخادم وتغليف الرسائل.

server-stubs: يستقبل الرسائل، ويفكك الرسائل المجمعة، وينفذ الإجراءات على الخادم.

- تمثيل البيانات في جهاز المرسل والمستقبل بطريقتين:

١. big-endian

٢. little-endian