Processes

اللَّهُمَّ صَلِّ عَلَى مُحَمَّدٍ وَعَلَى آلِ مُحَمَّدٍ، كَمَا صَلَّيْتَ عَلَى إِبْرَاهِيمَ، وَبَارِكْ عَلَى مُحَمَّدٍ وَعَلَى آلِ مُحَمَّد،
كَمَا بَارَكْتَ عَلَى آلِ إِبْرَاهِيمَ، فِي الْعَالَمِينَ، إِنَّكَ حَمِيدٌ مَجِيدٌ.

By: Mohamed Gamal Maklad

❖ **Process**:

- Also known **as job**.
- **process is a program in execution**. طول ما البرنامج في الميوري فهو اسمه برنامج اول ما نعمل عليه عمليات يتجول اسمه الي **Process**
- **Process execution must be sequential**. العمليات لازم تتنفذ بالترتيب اللي هو يعني علي التوالي
- Process current status is represented by: حالة العمليه بتتحدد بعاملين
  - **The value of the program counter** (***PC***) ده بيكون فيه رقم العمليه اللي عليها الدور في التنفيذ
  - The **contents of the processor's registers**.

Execute ***ولا decode*** ***ولا fetch*** *هل هي بتعمل* ***Register*** *محتوي ال الخاص بالعمليه*

❖ **Process Content**:

- ***Text section***: executable code. دي بيتخزن فيه الأوامر اللي هتتنفذ علي العمليه
- ***Data section***: global variables. دي بتكون القيم اللي بتكون متعرفه من اول ما العمليه تشتغل لجد ما تخلص و تقفل
- **Heap section**: dynamically allocated variables. لو معرف اوبجيكت مثلا فقيمته هتتخزن فيه انما العنوان بتاعه بيتخزن في الاستاك
- ***Stack section***: temporary variables (local variables) بيكون متغير هتاجه مثلا داخل ميثود و اول ما تخلص خلاص مش هحتاجه

❖ **Process States**: حالة العمليه دي بتكون ايه تعالي يلا نشوف الأنواع بتاعتها

- ***New***. The process **is being created**. دي اول ما العمليه تتعمل
- ***Running***. Instructions **are being executed**. اثناء تنفيذ أوامر عليها
- ***Waiting***. The process is **waiting for some event to occur**. لما تكون مستنيه حاجه تحصل زي مثلا انك تدخل رقم من الكيبورد
- ***Ready***. The process is **waiting to be assigned to a processor**. هنا جاهزه للتنفيذ مستنيه البروسيسور ياخدها
- ***Terminated***. The process has **finished execution**. خلاص خلصت اللي مفروض تعمله

New ➔ Ready➔ Running ➔ Terminated

بص ي هندسه صل علي النبي الأول اول ما العمليه تكون في وضع **Running** لو حصلها **Interrupt** بترجع وضع ال **Ready** طب لو كانت مستنيه **I/O** تخش في وضع **Wait** الحمدلله **I/O** وصل بالسلامه تقوم تخش وضع **Ready** وبعد كده تستني يتعملها **Running**

❖ **Process Control Block (PCB)**:

- Also know as **task control block**.
- Each process **is represented in the OS by a PCB**.
- ***PCB*** **data structure that stores information about a process**.
- **OS uses a PCB to store all the data needed to start** (or restart) **a process, along with accounting data**

شايف وانت جاي الصبح الكليه ساعه 8 و مزاجك وحش جيت تعدي من البوابه الامن قالك الكارنيه ي بشمهندس علشان اخليك تعدي اهو برضه الprocess من غير PCB اللي هو بالنسبالها زي الكارنيه OS مش هتخليها تعدي لان ده بيكون فيه كل المعلومات عن العمليه

❖ **PCB Contents**: ايه بقي المحتوي بتاعتها او البيانات المهم اوي دي اللي موجوده جواها

- ***Process state***. (e.g. new, ready, etc.). حالة العمليه دي **ايه هل هتتنفذ ولا هتستني**
- ***Program counter***. The address of **the next instruction**. عنوان الامر اللي عليه الدور في التنفيذ
- ***CPU registers***. **Stores state information when an interrupt occurs**. ده لو حصل مقاطعه بيخزن الحاله اللي كانت عليها
- ***CPU-scheduling information***. Scheduling parameters (e.g. priority). هنفهمها قدام
- ***Memory-management*** information. Process location in memory مكان العمليه دي في الميموري

- ***Accounting information***. time limits, process numbers, and so on.اللي باقي ليها في الوقت المسموح دي اتشرحت في اول فايل
- ***I/O status information***. Allocated I/O devices, open files, and so on. بيحجز أجهزة الإخراج والادخال اللي هتحتجها

❖ **Process Scheduling**:

- ***CPU scheduler*** selects a **ready process from memory for execution**.بيختار العمليه اللي هتتنفذ من العمليات المستعده بس
- ***Degree of multiprogramming*** is the **number of processes in memory**.
- Processes can be described as either I/O-bound or CPU-bound.
  - ***An I/O-bound*** process ***spends more of its time doing I/O***. **دي اغلب وقتها بيكون مدخلات و مخرجات**
  - ***A CPU-bound*** process uses **more of its time doing computations معقده عمليات وقتها اغلب بقي دي**
- **Scheduling Queues**:
  - ***Ready*** queue contains **processes waiting to execute on a CPU**. **فيه العمليات اللي جاهزه تتنفذ**
  - ***Wait*** queue contains **processes waiting for a certain event to occur حاجه مستنيه اللي العمليات فيه متخزن بقي ده**

❖ **Context Switch**:

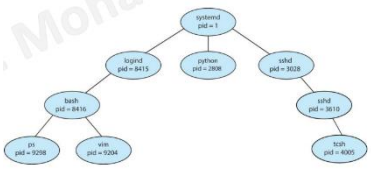
- The **context of a process is represented in its PCB**.
- **Context includes value of CPU registers, location in memory**, etc.
- **Context switch occurs when the CPU switches to another** process
- **Context switch**
  - ***Saves*** the **current context of the current process in its PCB**.
  - ***Loads*** the **saved context of the other process scheduled to run**.

ال context switch ان يكون في عمليه شغاله و اجي احب اشغل عمليه تانيه جه في بالك سؤال صح مجاش طب ماشي هقولك دلوقتي انت بتقول هبدل من عمليه 1 مثلا ل عمليه 2 طب لو رجعت ل عمليه 1 هبدأ تاني من الأول هقولك لا هما بقي بيخزنوا اللي العمليه كانت عملته في PCB بتاعتها بعمليه اسمها Save ولما يجي يرجع يكمل بيعمل load او تحميل للي كان وصله

- Context switch time **is pure overhead** (no useful work while switching) **لما اجي ابدل من عمليه للتانيه في وقت بيضيع**

❖ **Process Creation**:

- A **process may create several new processes البروسيس ممكن تعمل أي عدد من البروسير التابعه ليها**
- The **creating process is called a *parent* process**.
- The **new processes are called the children of that process**.
- **Each of these new processes may, in turn, create other processes**



❖ **Parent-Child Process Relationship**: العلاقه بينهم بقي

- In terms **of execution, there are two possibilities**: في حالة التنفيذ في احتمالين
  - **parent continues to execute concurrently with its children. ان الاب و الأطفال يتنفذوا مع بعض**
  - **parent waits until some or all of its children have terminated. الاب يستني الأطفال يخلصوا الأول علشان مستني منهم حاجه اكيد**
- **In terms of address-space, there are two possibilities:المكان اللي هيتعمل فيهم الاطفال**
  - **The child process has a new program loaded into it. ممكن يكون في برنامج تاني**
  - **The child process is a duplicate of the parent process ممكن يكون في نفس البرنامج**

- In terms of **resource sharing, there are three possibilities**: مشاركة الموارد
  - **Parent and child share no resources**. مفيش مشاركه بينهم
  - **Parent and children share all resources.** بيشاركوا كل الموارد مع بعض
  - **Children share a subset of parent's resources** بيشاركوا جزء و جزء

❖ **Process Termination**:

- A **process terminates when it finishes execution** البروسيس بتقفل لما تخلص شغل
- **It asks the OS to delete it by using the exit system call** عن طريق انها بتتطلب من نظام التشغيل انها يخرجها
- Once a **process is terminated, the OS deallocates all of its resource** خلاص خلصت بدمر كل مواردها معنتش محتاجها
- A **child process may output data to its parent using *wait system call***

❖ **Child Process Termination**:

- **parent process may terminate its child using the *abort* system call**.
- **This may happen in the following cases:**
  - **Task** assigned to the **child is no longer required.** خلاص معنتش محتاج الطفل ده دوره خلص
  - **Parent is exiting** in a **cascading termination** system.الاب خلص اللي مفروض يعمله ف كل اطفاله لازم يتقفلوا
  - **Child has exceeded the usage of resources allocated to it.** بيستخدموا موارد زياده عن اللزوم
- ***Cascading termination*** means that if **a process terminates, all of its children must be terminated as well**

بيقولك يعني لو في عمليه خلصت خلاص لازم كل الأطفال اللي هي عملاهم يتقفلوا برضه

❖ **Cooperating Processes**:

- ***independent*** process **does not share data with other processes.**دي انانيه مبتشاركش الداتا مع حد
- ***cooperating*** process **shares data with other executing processes**. اما دي بقي بتشارك الداتا مع باقي العمليات
- **Reasons to support process cooperation**:
  - ***Information sharing***. (e.g. copying and pasting).
  - ***Computation speedup*****. Break a task into sub-tasks that run in parallel**. مهو لما يكون في اكتر من عمليه في بينهم

    تواصل طب ما انا ممكن استغل ده و اقسم الاسك اللي عندي عليهم وده هيوفر عندي الوقت و هيخليني انجز بسرعه
  - ***Modularity***. Dividing the functions of a system **into separate processes.**زي ما قولنا لو نظام ضخم بقسمها لأجزاء

- cooperating processes need an IPC mechanism to exchange data. That may be: shared **memory or message passing**.

❖ **Shared Memory**: العمليتين لازم علي نفس الجهاز

- Normally, the **OS restricts a process to its own memory**.
- **Communicating processes agree to remove this restriction**.
- **process creates a shared-memory region in its address space**.
- Other processes can then attach that region to their address space.
- They **exchange information by reading and writing data in that region**
- Communicating processes are responsible for:
  - **Determining the form of the data and the location of shared data**.

بص هو الطبيعي ان OS بتمنع أي process لنها تكتب عند التانيه في الحاله دي في اتنين process واحده بتخصص مكان عندها للكتابه بيحيث واحده تكتب و التانيه تقرأ ويتواصله في المكان ده يجي بقي kernel مليش دعوه بقي بيكم مش هتمشوا بدماغكم انتم مسئولين انكم متعملوش خطأ ولو حصل انتم اللي مسئولين عنه

- **Ensuring that they are not writing to the same location simultaneously لازم يتأكدوا ان العمليتين ميكتبوش ف نفس الوقت او واحده تقرأ قبل ما التانيه تكتب لازم واحده تكتب ثم بعد كده التانيه تيجي تقرأ**

❖ **Message Passing: ده بقي لو عاقو اتنين علي نفس الجهاز او جهازين مختلفين يتواصلوا مع بعض**

- **Allow processes to communicate without sharing memory.**
- **Useful in a distributed environment and managed by the OS. هنا بقي نظام التشغيل اللي بينظم الدنيا**
- A message-passing facility should provide at **least** two operations:***send***$(message)$, ***receive***$(message)$
- **Message Size**:
  - ***Fixed-sized messages*:ده حجم محدد للرساله طب لو عاوز ابعت رساله اكبر من الحجم ده يقولك مليش دعوه قسم**
    - **Easier to implement at the system-level**
    - **Place a restriction on the application programmer.**
  - **Variable-sized massages: هنا أي حجم هياخده مش محدد حجم معين**
    - **More complex to implement at the system-level**
    - **Makes application programming easier and more flexible**

❖ **Communication Link**:

- ***physical implementation*** of a link can take several forms As **hardware bus, network, ..**etc
- ***logical implementation*** **of a link can take several forms as well.**
  - **Direct or indirect** communication
  - **Synchronous** or **asynchronous** communication
  - **Automatic or explicit buffering**

❖ **Direct Communication**: أنواع التوصيل المباشر وايه الفرق بينهم

- **symmetric addressing**, both sender and receiver name each other. **هنا بيحدد هيبعت لمين و هيستلم من مين**
  - ***send***$(P, message)$: **Send a message to process P.**
  - ***receive***$(Q, message)$: **Receive a message from process Q.**
- ***asymmetric addressing***, only the sender names the receiver process. **هنا بحدد بس هبعت لمين**
  - ***send***$(P, message)$: **Send a message to process P.**
  - ***receive***$(id, message)$: Receive a message from any process. **هنا بقي مستني زي اللي يبعت استلم منه**
- **direct communication link has the following properties**:
  - **A link is established automatically**.
  - **A link is associated with exactly two processes**.
  - **Between each pair of processes, there exists exactly one link**.

❖ **Indirect Communication: هنا زي مثلا عمليه بتبعت ل صندوق البريد و التانيه بتروح تجيب من صندوق البريد مفيش تعامل مباشر**

- Messages are sent to and received **from mailboxes (or ports)**
- ***Mailbox* user** is a process **that can send messages to a mailbox**.
- ***Mailbox owner*** is a process **that can receive messages from mailbox**.
- In **indirect communication, *send*() and *receive*() are defined as**:

- *send*$(A, message)$: **Send a message to mailbox** A.
- *receive*$(A, message)$: **Receive a message from mailbox A.**

➢ **indirect communication link has the following properties**:

- A link is established by **creating and sharing a mailbox**.
- A link may be **associated with more than two processes**.
- **Each pair of processes may have several links (i.e. mailboxes).**

❖ **Synchronization**:

➢ ***Blocking (synchronous) send***. The sending **process is blocked until the message is received by the receiving process or by the mailbox.** هنا بيعت الرساله ويفضل مستني لحد ما يتم تسليمها و بعدين يكمب شغله

➢ ***Non-blocking (asynchronous) send***. The **sending process sends the message and resumes operation, immediately.** هنا بيعت الرساله و يكمل شغله علطول

➢ ***Blocking (synchronous) receive***. The **receiver blocks until a message is available. It resumes operation only after receiving the message.** يفضل واقف مستني مبيعملش حاجه لحد ما يستلم الرساله وبعد كده يشتغل

➢ ***Non-blocking (asynchronous) receive***. The **receiver retrieves either a valid message or a null and resumes operation, immediately.** في رسايل مبعوته يستلمها ماشي مفيش بيشتغل

❖ **Buffering**: مساحه لاستلام الرسايل اللي بتتبعت دي

➢ **Zero capacity (no buffering)** هنا الراسل بيعمل بلوك وميبعتش حاجه لحد ما الرساله الواحده يتم استلامها

- The link cannot have any messages waiting in it.
- The **sender must block until the recipient receives the message**.

➢ **Bounded capacity (automatic buffering)** هنا الرسايل اللي بتتبعت بتتخزن في كيو لو اتملي يعمل بلوك وميبعتش الا لما يفضي

- The queue has **finite length n; thus, at most n messages can reside in it**.
- If **the queue is full, the sender must block until there is an available space**.

➢ **Unbounded capacity (automatic buffering)** هنا ابعت براحتك المساحه اللي مخصصها لتخزين الرسايل كبيره وتعتبر ما لا نهايه

- The **queue's length is infinite; thus, any number of messages can wait in it**.
- **The sender never blocks**.

Client-Server Communication:

❖ **Sockets**:

➢ A **socket is an endpoint for communication**.

➢ **Identified** by **an IP address** concatenated with **a port number**.

➢ A **pair of processes communicating employs a pair of sockets**.

➢ The **server waits for client requests by listening to a specified port**.

➢ **request received, server accepts a connection from client socket**.

علشان اتنين process يتواصلوا مع بعض لازم Socket بس كده هقولك لا لو قولتلك اتصل عليا هتقولي مش معايا رقمك هنا نفس الحكايه لازم كل process تكون عارفه رقم التانيه اللي هو IP هتقولي تمام رقمك معايا اكلمك واتس ولا تليجرام هنا برضه نفس الحكايه لازم تحدد المكان بالظبط كمان اللي هو اسمه port number

❖ **Remote Procedure Call (RPC)**:

- **Allows a client to invoke a procedure on a remote host**.
- **RPC** daemon listens to a port on the remote system.
- **stub** on the client-side hides the communication details.
- ***stub*** allows **for calling the remote procedure like a local one**.
- **RPCs may fail or execute multiple times due to network errors**.
- **Messages exchanged in RPC communication are well structured**.
- **message contains the function identifier and passes parameters**.
- **function is executed, and any output is returned as a message**.

عندنا مثلا سيرفير عاملين عليه وليكن ميثود بتجمع رقمين بس الميثود دي مزيفه يعني ايه يعني كل اللي بتعمليه انها تاخد parameter مني اليوزر وبعد كده تبعته علي ميثود في مكان تاني ده اسمه *Stub* الميثود اللي في مكان تاني دي هي اللي بالفعل اللي بتعمل العمليه الحسابيه فكده انا استخدمت ميثود في مكان بعيد عملت *RPC*

يبقي ايه اللي بيحصل انا كمبرمج بستخدم Stub بتروح تنادي علي RPC وتباصي ليها parameter وبعد كده تاخد منها النتائج و ترجعهالي

اللَّهُمَّ صَلِّ عَلَى مُحَمَّدٍ وَعَلَى آلِ مُحَمَّدٍ، كَمَا صَلَّيْتَ عَلَى إِبْرَاهِيمَ، وَبَارِكْ عَلَى مُحَمَّدٍ وَعَلَى آلِ مُحَمَّد،
كَمَا بَارَكْتَ عَلَى آلِ إِبْرَاهِيمَ، فِي الْعَالَمِينَ، إِنَّكَ حَمِيدٌ مَجِيدٌ.

By: Mohamed Gamal Maklad