CH3: Processes

⇐ لما بدنا نعمل برنامج، أول إشي بنكتبه بلغة برمجية مثل ال C، ++C (يعني بنكتبه ب High level language)، والكمبيوترات ما بفهموا هاي اللغة، فالكومبايلر بحوّل الكود لـ machine code عشان الجهاز يفهمه، وعشان البرنامج يتنفّذ لازم هاد الكود يتم تحميله على الميموري ويلزمه بعض الريسورسز اي الأدوات بوفرّها

* Process Concept (Job = process)

process: program in execution

- Text section: The program code
- Program counter: A Register has the address of the next instruction
- stack: containing temporary data
- Data section: containing global variables
- Heap: containing memory dynamically allocated during run time

* Program is a passive entity stored on disk (executablefile) while the process is active

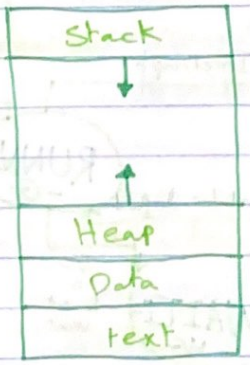
⇐ البرنامج هو جزء غير فعّال في الميموري لحدّ ما يتحوّل لبروسس (يصير تحت التنفيذ)

⇐ عشان نشغّل البرنامج إما نضغط عليه بالماوس أو عن طريق ال command line

. One program can be several processes

⇐ الجهاز قادر على تنفيذ أكثر من برنامج بنفس الوقت، والبرنامج الواحد ممكن يحتوي على أكثر من بروسس

* Process in Memory



⇐ طبعاً في مساحة بين ال Heap وال Stack عشان احنا ما بنقدر نحدد كم بدهم مساحة بالزبط لهيك بتكون dynamic

STUDENTS-HUB.com Uploaded By: Malak Obaid

* Process state

⇐ لما البروسس يتم تنفيذها حالتها بتتغير

Process state: The current activity of that process

NEW The process is being created.

⇐ لما يتم إنشاء البروسس.

RUNNING Instructions are being executed.

⇐ لما الإنستركشنز الي جوا البروسس عمّالهم بتنفذوا.

WAITING The process is waiting for some event to occur.

⇐ لما تكون بتستنى باشي يصير مثل الـ I/O أو تكون تستنى بسيغنال تيجيها.

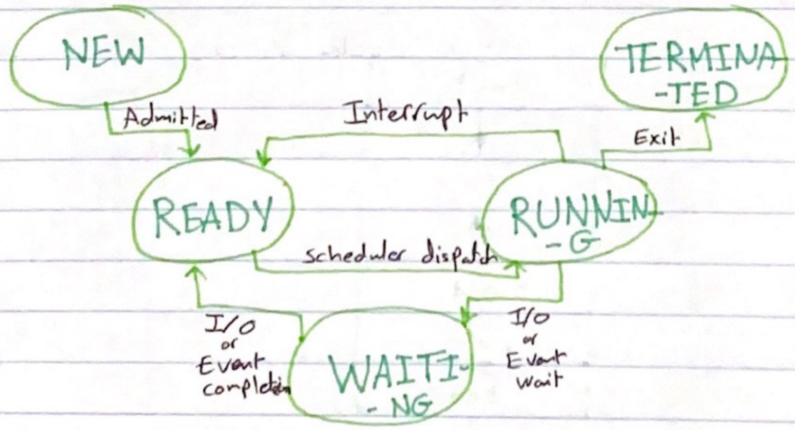
READY The process is waiting to be assigned to processor.

⇐ لما تكون جاهزة وبتستنى إنها تروح على البروسسر عشان تنفذ

TERMINATE The process has finished execution.

⇐ لما البروسس تخلّص التنفيذ تاعها

* Diagram of Process state



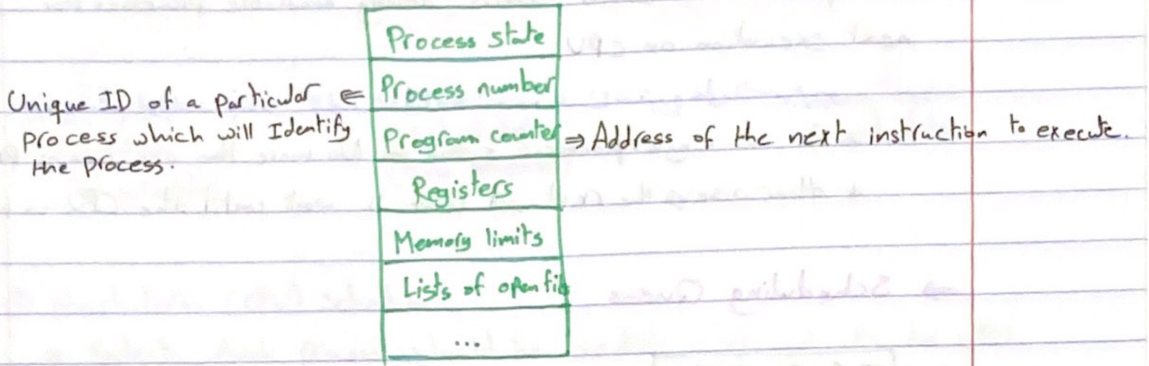
⇐ لما ننشئ البروسس بتصير New بعدها بتصير Ready وبتستنى تروح على البروسسر، فبعدها لما تروح على البروسس بتصير Running يعني بتصير تنفذ، بعدها في أكثر من احتمال أو حالة، الأولى إنها تجهز وتخلصها وتصير terminated، الحالة الثانية إذا صار interrupt يعني إذا اجت بروسس أهم منها بتروح على الـ Ready كمان مرة، الحالة الثالثة، إذا كانت بحاجة إنها تستنى I/O أو event معينة بتروح تصير waiting لحد م هاي الـ event تكتمل أو تجهز بتروح على الـ Ready وهكذا.

STUDENTS-HUB.com Uploaded By: Malak Obaid

* Process Control Block (PCB)

⇐ كل عملية بتمّ تمثيلها بالـPCB

. It also called a task control block.



Process state
Process number
Program counter
Registers
Memory limits
Lists of open fil
...
Unique ID of a particular process which will Identify the process.
Address of the next instruction to execute.

- CPU Registers: The registers that are being used by a particular process
- CPU scheduling information: has The priority of the processes it has the pointer to the scheduling queue and also other scheduling parameters.

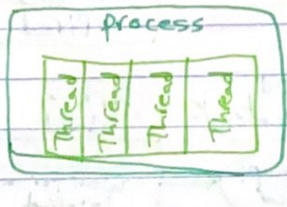
⇐ بكون في عدة بروسسز بدها تنفذ وبتستنى عشان تنفذ فالـscheduling بحدد أي عملية لازم تنفذ في الأول أو بتحدد ترتيب تنفيذ البروسسز

- Memory management information: represents the memory that is being used by a particular process
- Accounting information: It keeps an account of certain things like the resources that are being used by the particular process (CPU, time, memory, etc.)
- I/O status information: represents the I/O devices are being assigned to a particular process

* Threads

Thread: The unit of execution within a process.

⇐ البروسس ممكن تحتوي ثريد أو أكثر موجودة



* Process scheduling

⟵ بنظّم العمليات عشان نحقق أعلى استفادة من ال CPU، وعشان نسرّع ال switch بين البروسس

- The process scheduler selects among available processes for next execution on CPU

⟸ اي بنظم العمليات بختار العمليات المتاحة للتنفيذ عشان تروح على السي بي يو.

* For a single-processor system ⟹ No more than one running process
* otherwise ⟹ the rest will have to wait until the CPU is free

⟹ Scheduling Queus

$\boxed{\text{JOB QUEUE}}$ Set of all processes in the system.

⟸ لما البروسس تدخل على السستم بتم إضافتها على هاي الكيو.

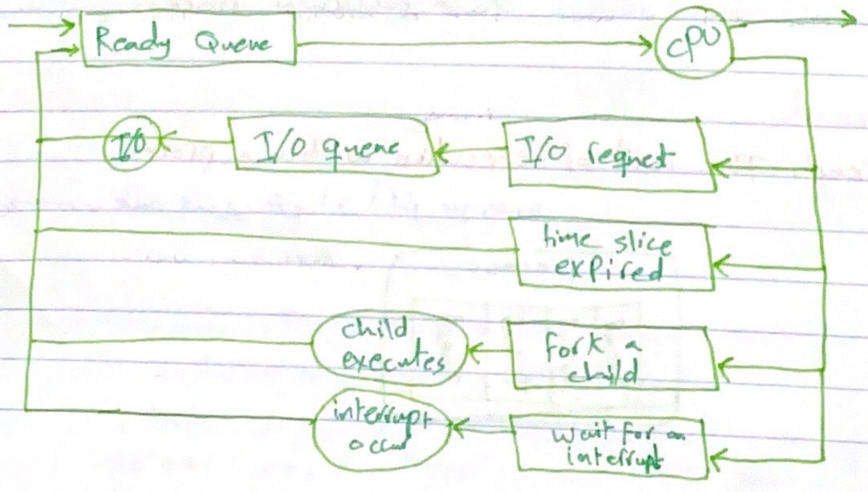
$\boxed{\text{READY QUEUE}}$ set of all processes residing in main memory, ready and waiting to execute.

⟸ البروسسز اللي بكونوا جاهزين وموجودين في الميموري عشان يتم تنفيذهم.

$\boxed{\text{DEVICE QUEUE}}$ set of processes waiting for an I/O device

⟸ مجموعة البروسسز اللي بكونوا يستنوا بأشي من ال I/O

* Representing of Process Scheduling



Uploaded By: Malak Obaid

* Schedulers

$$\begin{tikzcd}
 & \text{Schedulers} \arrow[dl] \arrow[d] \arrow[dr] & \\
\text{Short-term scheduler (CPU scheduler)} & \text{Medium-term scheduler (Job scheduler)} & \text{long-term scheduler}
\end{tikzcd}$$

① Short-term (CPU) scheduler

* Selects which process should be executed next and allocates CPU.

⇐ بحدد شو العملية الي لازم تتنفذ بعد العملية الي في السي بي يو.

* Sometimes the only scheduler in a system.

* must be fast (milliseconds).

② Medium-term scheduler

* Can be added if degree of multiple programming needs to decrease.

⇐ هو شيء اختياري بس بنقدر نضيفه عشان نقلل الـ multiple prog. ويكون أقل عبء على الجهاز

* Remove process from memory, store in disk, bring back in from disk to continue execution : swapping.

⇐ بشيل العملية من الميموري، وبخزنها على ديسك، وبرجع بجيبها من هناك عشان أكمل التنفيذ.

③ long-term scheduler (Job scheduler)

* Selects which processes should be brought into the ready queue.

⇐ بحدد شو العمليات الي لازم نوديها على الـ ready queue.

* May be slow (seconds, minutes).

* It controls the degree of multiprogramming.

$$\begin{tikzcd}
 & \text{I/O-bound process} \Rightarrow \text{more time } \boxed{\text{I/O}}\text{, less } \boxed{\text{computations}} \\
 & \text{* short CPU bursts.} \\
\text{Processes} \arrow[uur] \arrow[ddr] & \\
 & \\
 & \text{CPU\_bound process} \Rightarrow \text{more time } \boxed{\text{Computations}} \\
 & \text{* few very long CPU bursts.}
\end{tikzcd}$$

* long-term scheduler strives for good process mix.

STUDENTS-HUB.com Uploaded By: Malak Obaid

* Multitasking in Mobile Systems

⇐ في بعض الموبايلات بتسمح بس لبروسس وحدة إنها تكون شغالة.

① IOS

↳ Single foreground process - controlled via user interface
↳ Multiple background processes - in memory, running but not on the display, and with limits.

⇐ في قسمين بالنسبة للـ iOS القسم الأول الي هو العملية على الواجهة، بتكون بس وحدة، والقسم الثاني الي هي العمليات في الخلفية زي العموري بس بتكون محدودة.

* limits on background processes: single, short task, receiving notifications, long-running like audio

⇐ بتكون محدودة ومناصبة إنها تكون قصيرة زي استلام الإشعارات أو طويلة زي الأغاني مثلاً.

② Android

* Runs foreground and background, with fewer limits.
* Background uses a $\boxed{\text{Service}}$ to perform tasks.
  - ↳ Can keep running even if b.g. process is susbended
  - ↳ has no UI, small memory.

⇐ الأندرويد بشغل البروسس الي بالخلفية والي عالشاشة، الي بالخلفية بستخدم شي اسمه service ومن إيجابياته إنه بضل شغال حتى لو البروسس علقت.

* context switch

⇐ لما يصير إنتربت، السستم لازم يحفظ معلومات العملية الي بيعمل فيها وبنفذ فيها عشان يقدر يسترجعها بعد ما يخلّص الـ Interrupt (عشان يقدر يعرف وين وصل بتنفيذ هاي البروسس).

* Context switch: Performing a state save of the current process and a state restore of a different process.
* Context of a process represented in the PCB.

⇐ يعني معناها باختصار هو التبديل من عملية لعملية مع حفظ معلومات العملية الأولى عشان نرجع نكمل فيها بعد ما تخلص العملية الثانية.

* context switch is overhead (بشكل عبء لأنه النظام ما بعمل اشي أثناء هاي العملية)

⇐ أحياناً بوخذ وقت (بعتمد على طبيعة الهاردوير للجهاز)

* Some hardware provides multiple sets of registers per CPU
  ⟹ multiple contexts loaded at once.

STUDENTS-HUB.com
Uploaded By: Malak Obaid

* Operations on processes

- process creation
- process termination

* Process creation

* البروسس الوحدة ممكن تنشئ عدة بروسس أثناء التنفيذ، ولكل بروسس PID خاص فيها.

- parent process: The creating process.
- children process: The new processes.

⇒ Parent process create children processes which in turn create other processes, forming a tree of processes.

* Process identified and managed Via process identifier (PID)

⇒ Resource sharing options:

a. Parent & children share all resources ⇐ (بتشاركوا بكل الريسورس)

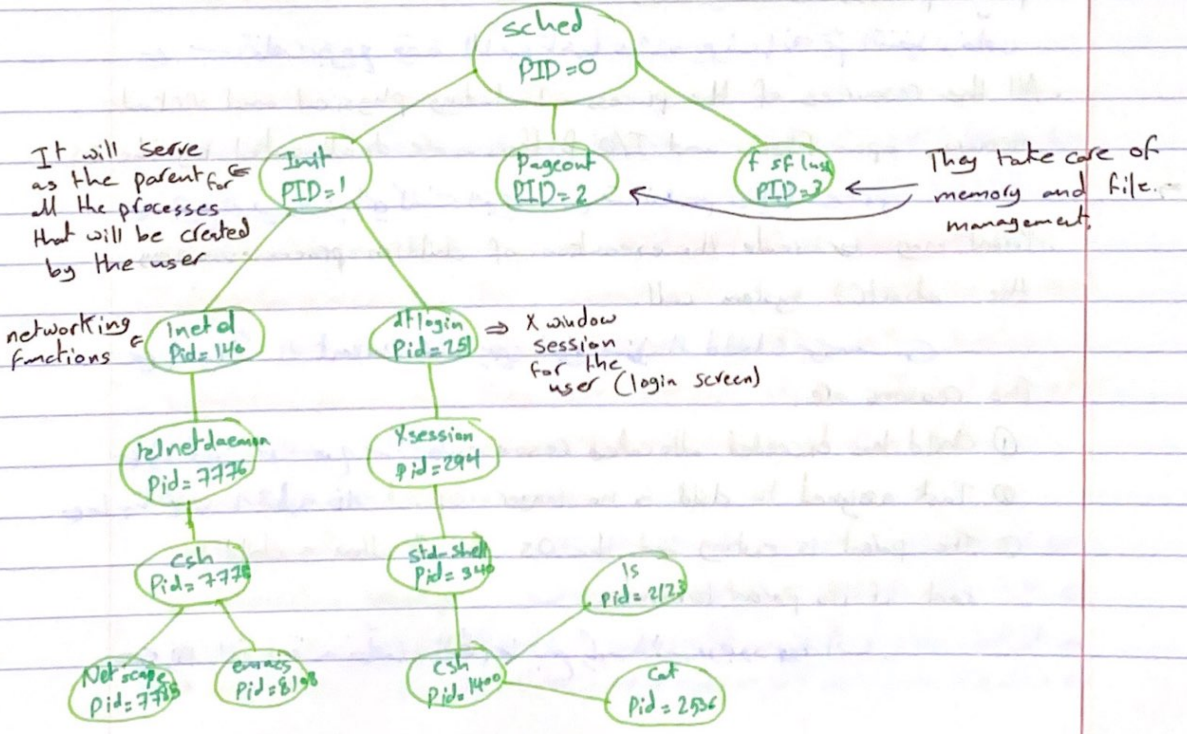
b. children share subset of parent's resource. ⇐ (الأبناء بتشاركوا مع الأب في بعض الريسورس)

c. Parent & children share no resources ⇐ (ما بتشاركوا بالريسورس بالمرة)

⇒ Execution options:

a. Execute concurrently. ⇐ بتم تنفيذهم سوا مع بعض.

b. Parent wait until children terminate. ⇐ الأب بستنى لحد ما يخلصوا الأبناء



⇒ Address space :

a. child duplicate of parent. ⇐ الـ child عنده نفس البرنامج والداتا الخاصة بأبوه.

b. child has a program loaded into it. ⇐ الابن عنده برنامج ثاني لازم ينفذه

• UNIX examples

. fork() ⇒ system call creates new process.

. exec() ⇒ system call to replace the process' memory space with a new program.

⇐ لما ننفذ الأمر fork() بترجعلنا قيمة الـ Pid ، إذا كانت بالسالب معناها العملية فشلت ، لما تكون رجعت صفر هاد معناه إنه هاد الابن ولما ترجع غير هيك معناه إنه هاد الأب .

* Process Termination

• A process terminates when it finishes executing it's final statement and asks the OS to delete it by using exit().

⇐ البروسس بتخلص بس تنفذ آخر جملة وبعدها بتطلب من نظام التشغيل إنه يحذفها باستخدام الـ exit() .

• At that point, the process may return a status value to its parent process (via wait())

⇐ بس تخلص بترجع قيمة للأب تاعها عشان يعرف إنه تم التنفيذ وخلص .

. All the resources of the process – Including physical and virtual memory, open files, and I/O Buffers – are deallocated by the OS

⇐ جميع الريسورز الي كانت البروسس تستخدمهم بصيروا free

. Parent may terminate the execution of children processes using the abort() system call.

⇐ أحياناً الـ parent علمه ينهي البروسيس الـ child بروسس

. The reasons are:

① child has exceeded allocated resources ⇐ إذا تجاوز الحد المسموح من الريسورز

② Task assigned to child is no longer required. ⇐ ما بدنا هاي العملية خلص

③ The parent is exiting and the OS doesn't allow a child to cont. if its parent terminates.

⇐ إذا الأب نفسه خلص والنظام ما بسمح يكون الابن موجود بدون أب .

STUDENTS-HUB.com Uploaded By: Malak Obaid

. Some OSs don't allow child to exists if its parent has terminated
⟹ All its children, grand children will be terminated (Cascading termination)
⇐ إذا في عملية إلها أبناء تم إنهاؤها معناها جميع أبنائها، أبناء أبنائها يتم إنهاؤهم (في بعض أنظمة التشغيل)
⇐ يتم بدء الإنهاء بواسطة نظام التشغيل.

. wait() : A system call that returns status information and the pid of the terminated child process. to its parent

. no parent waiting ⟹ process is a zombie
. parent terminated without wait() ⟹ process is an orphan

* Multiprocess Architecture - chrome Browser

. Many web browsers ran as single process
⟹ Entire browser can be crashed if just one tab was having trouble.
⇐ المتصفحات زمان كانت كأنها عملية واحدة، إذا بس تاب واحدة خربت أو صار فيها مشكلة جميع المتصفح بتخرب (بصير فيه مشكلة).

. Google chrome is multiprocess browser
↳ Browser process ⟹ UI, disk and network I/O
↳ Renderer process ⟹ web pages, deal with HTML, JS
↳ Plug-in process ⟹ for each type of plug-in

* Interprocess Communication

⇐ معناها الطريقة اللي البروسس بتواصلوا مع بعضهم، لأنه أحياناً العمليات بتحتاج إنها تتواصل مع بعضها.

. Processes within a system may be independent or cooperating.
. Independent processes : They cannot affect or be affected by each other processes executing in the system.
. Cooperating processes : They can affect or be affected by other processes, including sharing data.

. Reasons for cooperating processes:
⇒ Information sharing ⇐ ممكن عدة عمليات تكون بحاجة لملف معين بنفس الوقت
⇒ Computation speedup ⇐ لما نقسم التاسك لعدة تاسكات بتشتغل بنفس الوقت
⇒ Modularity ⇐ لما الجهاز ينصنع يكون أكثر من module ضمن الـ modules لازم يتواصلوا سوا
⇒ Convenience ⇐ مشان لما المستخدم يشغل أكثر من عملية جميعهم يشتغلوا بسلاسة

. Cooperating processes need Interprocess communication (IPC) mechanism that will allow them to exchange data & information

⇐ لازم يكون في اشي اسمه IPC بسمح للعمليات إنهم يتبادلوا البيانات والمعلومات

$$\begin{tikzcd}
 & \text{models of IPC} \arrow[dl] \arrow[dr] & \\
\text{shared Memory} & & \text{Message passing}
\end{tikzcd}$$

① Shared Memory

↳ A region of memory that is shared by cooperating processes is established.

↳ Processes can exchange info by reading & writing data to the region

⇐ بتم تأسيس منطقة من الميموري بتكون مشتركة بين العمليات الي بتتعاون وبتبادلوا المعلومات عن طريق الكتابة والقراءة على المنطقة المشتركة.

② Message passing

↳ communication takes place by means of messages exchanged between the cooperating processes

⇐ بكون في مسجات بتم تبادلها بين العمليات عن طريق الكيرنال.

* Interprocess Communication – shared Memory ⟵ (يعني راني مستبقة)

. Shared memory : An area of memory shared among the processes that wish to communicate

. The OS itself doesn't interfere in controlling the shared memory

⇐ نظام التشغيل ما بتدخل بالمنطقة هاي بين البروسيسز

. The major issue : processes cannot synchronize their actions when they access shared memory.

⇐ المعلومات ما بتم تزامنها لهيك ممكن يكون في خلل أو يصير تضارب في البيانات

* Producer-Consumer Problem

Producer process produces information that is consumed by a consumer process ☺?

⇐ المشكلة هون إنه لازم نخلي المنتج والمستهلك إنهم يشتغلوا سوا بنفس الوقت يعني المنتج رح ينتج والمستهلك رح يستهلك، ولازم يشتغلوا سوا عشان المستهلك بس يستهلك اللي بتم إنتاجه بس وما يحاول يستهلك اللي ما تم إنتاجه عشان هيك لازم يشتغلوا سوا.

① . One solution is to make shared-memory ‼ *معلش أنا كاتبة ما فهمت وين المشكلة ☺

②. To allow producer and consumer work concurrently, we must have available a buffer of items that can filled by the producer and emptied by the consumer.

⇐ الحل للمشكلة إنه يكون فيه buffer عشان المنتج يعبيه والمستهلك يفضيه ويكون ال buffer موجود في منطقة مشتركة بين المنتج والمستهلك.

Two kinds of buffers

- Unbounded buffer: no limit on the size of the buffer
- Bounded buffer: There is a fixed buffer size

* Message passing

↳ Mechanism for processes to communicate and to synch. their actions.

⇐ الطريقة اللي بتواصلوا فيها البروسسز بدون ما يستخدموا سبيس مشتركة طبعاً هاي الطريقة أفضل في حالة كانت الأجهزة مختلفة وفي حالة كان السستم أو الهاردوير مش مرتب بطريقة تسمح إنه يتعمل shared memory بسهولة.

• processes communicate with each other without resorting to shared variables.

* IPC facility provides two operations

⇒ send (message) ⇒ receive (message)

* Message size can be either fixed or Variable.
* If processes P and Q wish to communicate, they need to:
  1. Establish communication link between them
  2. Exchange messages via send/receive

⇒ Implementation of communication link:

- Physical:
  - shared memory
  - Hardware bus
  - Network
- logical:
  - Direct or indirect (Naming)
  - synchronous or asynchronous. (synchronization)
  - Automatic or explicit buffering. (Buffering)

* Direct Communication

⇐ بالتواصل المباشر كل بروسس لازم تذكر بشكل صريح اسم البروسس المستقبلة

- Process must name each other explicitly:
  - send (P, message) ⇐ ابعث الرسالة للبروسس P
  - receive (Q, message) ⇐ استقبل الرسالة من البروسس Q
- Properties of communication link:
  * links are established automatically. ⇐ يتم إنشاء اللينك بطريقة تلقائية
  * A link is associated with exactly one pair of communication processes
  * Between each pair there exists exactly one link
  * The link may be unidirectional, but it usually bi-directional.

* Indirect Communication

↳ Messages are directed and received from mailbox, or ports.

⇐ الرسائل بتنبعت للـmailbox وهناك بتم إرسالها للجهة المعنية وهكذا

⇐ كل mailbox عنده ID معين، والبروسس بقدروا يتواصلوا مع بعض بس إذا في mailbox مشترك بينهم

- Send (A, message) ⇐ ابعت مسج للـmailbox اللي اسمه A
- receive (A, message) ⇐ استقبل مسج من الـ Mailbox A

- properties of communication link
  - link established only if processes share a common mailbox
  - A link may be associated with many processes
  - Each pair may share several comm. links
  - link may be unidirectional or bi-directional

⇐ عملية التواصل غير المباشر: بعملوا mailbox جديد، بتبادلوا الرسائل من خلاله وبعدين بدمروه

⇐ صارت في مشكلة إذا أكثر من ٢ process بتشاركوا بنفس الـmailbox كيف بده يعرف مين الرسالة المعينة؟

- Solutions:
  1. Allow a link to be associated with at most two processes.
  2. Allow one process at a time to execute a recieve operation.
  3. Allow the system to select the receiver

* Synchronization

Message passing may be either blocking (synchronous) or nonblocking (Asynshronous).

⇐ تبادل الرسائل إما بكون متزامن أو غير متزامن.

- Blocking send: The sending process is blocked until the message is received. (synchronous)

⇐ ممنوع تبعت كمان رسائل لحد ما الرسالة توصل.

- nonBlocking send: The sending process sends the message and resumes operation. (Asynchronous)

⇐ مش لازم البروسيس تستنى المسج يوصل بتقدر تبعت كيف ما بدها.

- Blocking receive: The receiver blocks until a message is available

⇐ ممنوع تقبل أي رسالة لحد ما الرسالة الي بدها إياها توصل أو تنبعت.

- nonblocking receive: The receiver retrieves either a valid message or a null.

⇐ عادي تستقبل أي شي.

- rendezvous: send & receive are blocking.

STUDENTS-HUB.com Uploaded By: Malak Obaid

* Buffering

↳ Queue of message attached to the link

- → Zero capacity : no messages are queued on a link

  ⇐ المرسل لازم يستنى المستقبل (بس رسالة وحدة) بستنى الرسالة توصل عنده، ترجع يبعث

- → Bounded capacity : finite lenghth of n messages

  ⇐ المرسل لازم يستنى إذا اللينك فل (إذا عدد الرسائل بيساوي n)

- → Unbounded capacity : infinite length

  ⇐ المرسل ما بستنى وبضل "يبعث" بما أنه المساحة مش محدودة

* Communications in client-server systems (حصة ٦ عادة شفت الي قبل شوي)

- Sockets
- Remote Procedure calls
- Pipes
- Remote method Invocation

* Sockets

↳ Endpoint for communication

- A socket is identified by an IP address concatenated with a port number.

* السيرفر بستنى طلب الكلاينت عن طريق إنه ينتبه لـ port معين، فلما الطلب يوصل، السيرفر بقبل الاتصال ~~[illegible]~~ من السوكت الخاص بالكلاينت عشانه تتم عملية الاتصال.

* كل خدمة أو بروتوكول إلهم رقم Port معين خاص فيهم.

- All ports below 1024 are considered well-known

sockets
- → TCP
- → UDP
- → Multicast socket

*Remote Procedure Calls (RPC)

↳ Protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network's details

⇐ عبا [illegible] برنامج يطلب خدمة من برنامج على جهاز ثاني عن طريق الشبكة بدون فهم تفاصيلها.
طبعاً بتم استخدام ال ports في هاي الخاصية وهو

*Pipes

↳ Ordinary pipes

Cannot be accessed from outside the process that created it.

⇐ بتكون البروسس علاقتها بتتواصل مع الأبناء تاعونا.

↳ Named pipes

Can be accessed without parent-child relationship.