CH3: Processes

⇐ ما بنا نحل برنامج، أول شي بنكتبه بلغة برمجية مثل الـ C، C++ (يعني High level language)، والكمبيوترات ما بتفهم هاي اللغة، فالكومبايلر بحوّل الكود لـ machine code عشان الجهاز يفهمه، وعشان البرنامج يتنفّذ لازم هاد الكود يتم تحميله على الميموري وبلزمه بعض الريسورس اي الأوس بوفّرها

* 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 (executable file)
While the process is active

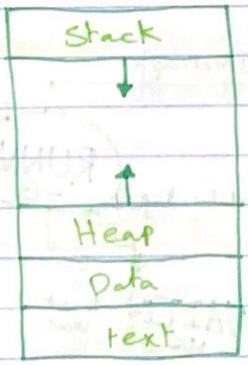
⇐ البرنامج هو جسم غير فعّال في الميموري لحتى ما يتحوّل لبروسس (بصير تحت التنفيذ)

⇐ عشان نشغّل البرنامج إما نضغط عليه بالماوس أو عن طريق الـ 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 الـ Heap والـ Stack عشان اذا ما بنقدر نحدد كم بدنا مساحة بالزبط ليلاء بتكون dynamic

STUDENTS-HUB.com Uploaded By: anonymous

* 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 عمالهم بتنفذوا.

WAITING → The process is waiting for some event to occur.

⇐ لما تكون بتستنى اشي يصير مثل الـ I/O أو تكون تستنى بسيجنال يجيها.

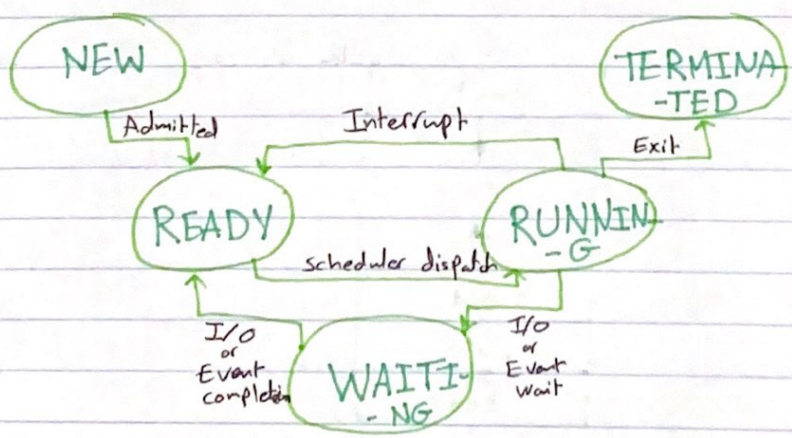
READY → The process is waiting to be assigned to processor.

⇐ لما تكون جاهزة وبتستنى إنها تروح على البروسسر عشان تنفذ

TERMINATED → The process has finished execution.

⇐ لما البروسس تخلص التنفيذ تاعها

* Diagram of Process state



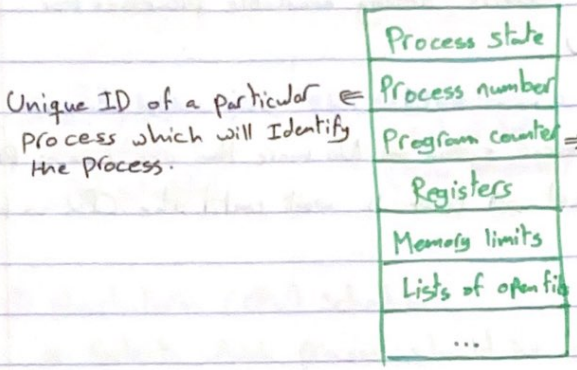
⇐ لما ننشئ البروسس بتصير New بعدها بتصير Ready وبتستنى تروح على البروسسر، فبعدها لما تروح على البروسسر بتصير Running يعني بتصير تنفذ، بعدها في أكثر من احتمال أو حالة، الأولى إنها تجهز وتخلصها وبتصير terminated، الحالة الثانية إذا صار interrupt يعني إذا اجت بروسس أهم منها بتروح على الـ Ready كمان مرة،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

STUDENTS-HUB.com Uploaded By: anonymous

* Process Control Block (PCB)

⇐ كل عملية بتتم تمثيلها بالـ PCB

. It also called a task control block.



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.

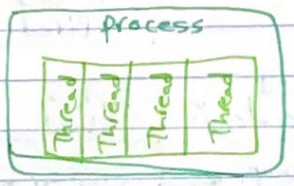
⇐ بكون في مدة بروسس بهما تنفذ وبتستنى شان تنفذ فال scheduling بحدد أي عملية لازم تنفذ في الأول أو بتحدد ترتيب تنفيذ البروسس

- 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 من واحدة



STUDENTS-HUB.com Uploaded By: anonymous

* Process Scheduling

← بنظم العمليات عشان نحقق أعلى استفادة من الـ CPU وعشان نوزعه

الـ switch بين البروسس

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

← اي بنظم العمليات بختار العمليات المتاحة للتنفيذ عشان نوزع 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 Queues

JOB QUEUE Set of all processes in the system.

← كل البروسس تدخل على السستم بيتم إضافتها على جوا الكيو.

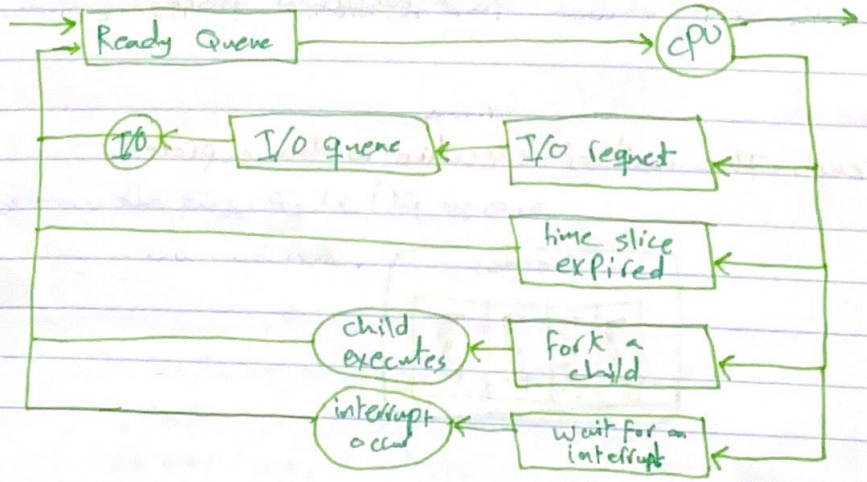
READY QUEUE set of all processes residing in main memory, ready and waiting to execute.

← البروسس اللي بكونوا جاهزين وموجودين في الميموري عشان يتم تنفيذهم

DEVICE QUEUE Set of processes waiting for an I/O device

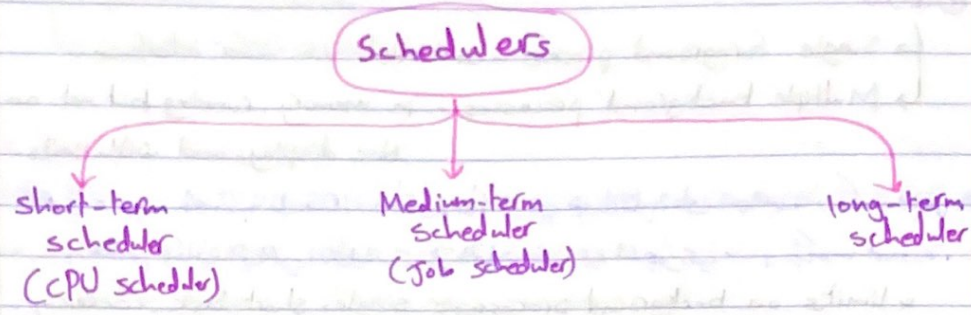
← مجموعة البروسس اللي بكونوا بيستنوا جهاز الـ I/O

* Representing of Process Scheduling



STUDENTS-HUB.com Uploaded By: anonymous

* Schedulers



① Short-term (CPU) scheduler

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

← يحدد أي العملية اللي لازم تنفذ بعد العملية اللي 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 multiprog. وكمان أقل عبء على الجهاز

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

← بشيل العملية من الميموري ويخزنها على ديسك ويرجع يجيبها من هناك عشان تكمل التنفيذ.

③ long-term scheduler (Job scheduler)

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

← يحدد العمليات اللي لازم نوزعها على الـ ready queue.

* May be slow (seconds, minutes).
* It controls the degree of multiprogramming.

Processes
- I/O-bound process ⇒ more time I/O, less computations
  * short CPU bursts.
- CPU-bound process ⇒ more time computations
  * few very long CPU bursts.

* long-term scheduler strives for good process mix.

STUDENTS-HUB.com Uploaded By: anonymous

* 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 العوري بس بتكون محدودة.

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

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

② Android

* Runs foreground and background, with fewer limits.
* Background uses a 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 عملية التبديل)

⇐ 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: anonymous

* Operations on processes
  - process creation
  - process termination

* Process creation

* العمليات الوحدة هكذا تنشئ عدة عمليات أثناء التنفيذ، وكل عملية لها PID خاص بها.

- 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 ← (يتشاركوا بكل الريسورس)

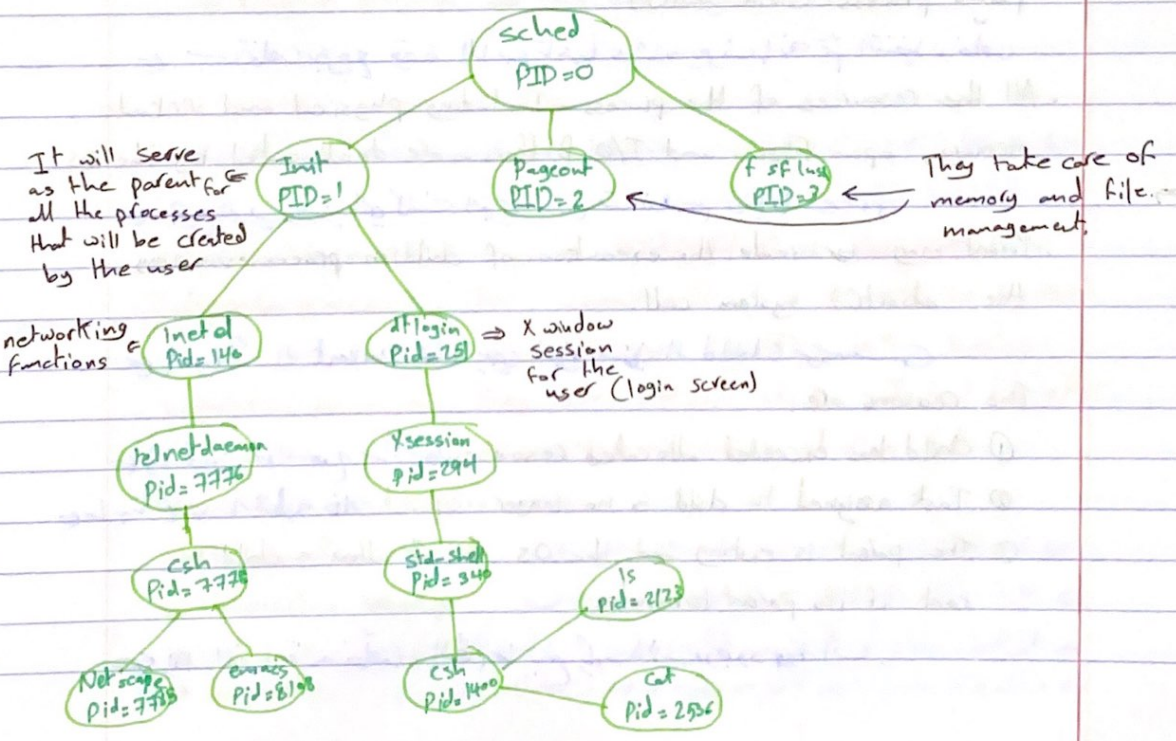
b. children share subset of parent's resource. ← (الأبناء يتشاركوا مع الأب في بعض الريسورس)

c. Parent & children share no resources ← (ما يتشاركوا بالريسورس بالمرة)

⇒ Execution options:

a. Execute concurrently. ← يتم تنفيذهم سوا مع بعض.

b. Parent wait until children terminate. ← الأب يستنى لين ما ينفذوا الأبناء



STUDENTS-HUB.com Uploaded By: anonymous

⇒ Address space:

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

b. child has a program loaded into it. ⇐ الابن عنده برنامج تاني 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 مشلت ، لما تكون رجعت صفر معناه إنه هاد الابن ولما ترجع غير صفر معناه إنه هاد الأب.

* Process Termination

• A process terminates when it finishes executing its 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 اللي كانت البروسس تستخدمها بيعملوا free

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

⇐ أحياناً الـ Parent عشان ينهي البروسس 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 بقا في حاجة للعملية خلص

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

⇐ إذا الأب بدو يخلص والنظام ما بسمح يكون الابن موجود بدون أب.

STUDENTS-HUB.com Uploaded By: anonymous

- 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 واحد خربت أو صار فيها مشكلة جميع المتصفح بكون (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 بعضهما، 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 على ملف معين بنفس الوقت
  ⇒ Computation speedup ⇐ لما نقسم التاسك لعدة تاسكات بتشتغل بنفس الوقت
  ⇒ Modularity ⇐ لما الجهاز مصنوع بكون أكثر من module ضمن الـ module لازم يتواصلوا سوا
  ⇒ Convenience ⇐ مثلاً لما المستخدم يشغل أكثر من عملية جميعهم يشتغلوا بسلاسة

STUDENTS-HUB.com Uploaded By: anonymous

- 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

models of IPC

- shared Memory
- Message passing

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

⇐ بيتم تأسيس منطقة من الميموري بتكون مشتركة بين العمليات اللي بتتشارك وبتبادلوا المعلومات من 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 ← (بعرف إني مسبقة)

- 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

STUDENTS-HUB.com Uploaded By: anonymous

* 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 كان 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 يعمل 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 المستقبل

- Process must name each other explicitly:
  - send (P, message) ⇐ ابعت 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 mailboxes or ports.

⇐ الرسائل بتتبعت للـ mailbox وهناك يتم إرسالها للجهة المعنية وهكذا

⇐ كل mailbox ليها IP معينة، والبروسس بقدروا يتواصلوا مع بعض بسهولة في mailbox مشترك بينهم

STUDENTS-HUB.com Uploaded By: anonymous

- send (A, message) ⇐ ابعت مسج للـ mailbox 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 الـ mailbox 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 (Asynchronous).

⇐ تبادل المسجات 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 اللي 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: anonymous

* 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 بعدين برجع يبعث
    - Bounded capacity: finite lenghth of n messages
      - المرسل لازم يستنى إذا اللينك فُل (إذا عدد الرسائل بيساوي n)
    - Unbounded capacity: infinite length
      - المرسل ما بستنى وبضل "يبعث" لأنه المساحة مش محدودة

* Communications in client-server systems (الحقة ناسد قت الي قبل شرحتا)
  - 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 بنتبه لـ port معين، فلما الطلب يوصل، السيرفر بقبل الاتصال من السوكت الخاص بالكلاينت عشان تتم عملية الاتصال.

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

- All ports below 1024 are considered well-known

sockets:
- TCP
- UDP
- Multicastsocket

* 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

← يا برنامج يطلب خدمة من برنامج على جهاز ثاني من طريق الشبكة بدون معرفة تفاصيلها.

طبعاً بيتم استخدام الـ Ports في هاي الحالة. وهيك

* Pipes

↳ Ordinary pipes

cannot be accessed from outside the process that created it.

← يعني البروسس عشان تستقبل مع الأبناء تاعونها.

↳ Named pipes

can be accessed without parent-child relationship.

STUDENTS-HUB.com Uploaded By: anonymous