Inter-Process Communication

We now look at how processes communicate amongst themselves

Inter-Process Communication

We now look at how processes communicate amongst themselves

Many processes can be created, process, then exit without needing to refer to any other process

Inter-Process Communication

We now look at how processes communicate amongst themselves

Many processes can be created, process, then exit without needing to refer to any other process

But there are many processes that need to send data to, or receive data from other running processes

Inter-Process Communication

We now look at how processes communicate amongst themselves

Many processes can be created, process, then exit without needing to refer to any other process

But there are many processes that need to send data to, or receive data from other running processes

For example, a new program starting might wish to tell the process managing the display that it wishes to pop up a window on the display

Inter-Process Communication

We now look at how processes communicate amongst themselves

Many processes can be created, process, then exit without needing to refer to any other process

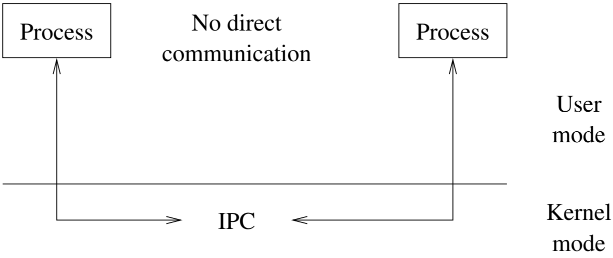
But there are many processes that need to send data to, or receive data from other running processes

For example, a new program starting might wish to tell the process managing the display that it wishes to pop up a window on the display

Or one process has to wait for another to finish some action (e.g., pop up a window) before it can progress itself: this is *synchronisation*

Inter-Process Communication

Inter-Process Communication (IPC) can be achieved in many different ways, but all must be, at base, supported by the OS; recall that by default the kernel tries to stop one process interfering with another



Inter-Process Communication

IPC contradicts this non-interference, and so must be treated very carefully by the kernel

Inter-Process Communication

IPC contradicts this non-interference, and so must be treated very carefully by the kernel

There must be rules and restrictions, or else one process could just blast another process with data, preventing it from doing any useful work

Inter-Process Communication

We shall be looking at

- Files
- Pipes
- Shared memory
- Signals
- Semaphores (synchronisation)
- Software buses

as a sample of IPC mechanisms

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

On the face of it this is just

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

On the face of it this is just

- Process A wishes to send some data to process B

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

On the face of it this is just

- Process A wishes to send some data to process B
- A writes it to a file

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

On the face of it this is just

- Process A wishes to send some data to process B
- A writes it to a file
- B reads it

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

On the face of it this is just

- Process A wishes to send some data to process B
- A writes it to a file
- B reads it

This seems easy

Inter-Process Communication

Files

A simple way for two processes to communicate is using an existing resource, namely files

On the face of it this is just

- Process A wishes to send some data to process B
- A writes it to a file
- B reads it

This seems easy

But it’s much harder than this

Inter-Process Communication

Files

- Which file to use? A and B need to agree on a filename to use, but this is not so easy. They can use a single “well-known” file, but this is problematic if many processes are all writing to the same file simultaneously. For example, C wants to communicate with D at the same time via the same file.

  They could have a separate file for each pair of processes, but to agree on a file name A and B must have previously communicated. . .

Inter-Process Communication

Files

- Which file to use? A and B need to agree on a filename to use, but this is not so easy. They can use a single “well-known” file, but this is problematic if many processes are all writing to the same file simultaneously. For example, C wants to communicate with D at the same time via the same file.

  They could have a separate file for each pair of processes, but to agree on a file name A and B must have previously communicated. . .
- How does B know when data has arrived? B might have to repeatedly poll the file until the data arrives. This doesn’t scale well to large numbers of files or processes

Inter-Process Communication

Files

- The file protections must be set properly (recall userids) to allow only the authorised processes to read/write to them

□

Inter-Process Communication

Files

- The file protections must be set properly (recall userids) to allow only the authorised processes to read/write to them
- Files are quite slow relative to the mechanisms we are going to see later

□

Inter-Process Communication

Files

- The file protections must be set properly (recall userids) to allow only the authorised processes to read/write to them
- Files are quite slow relative to the mechanisms we are going to see later

In general, files are not used for IPC

□

Inter-Process Communication

Files

- The file protections must be set properly (recall userids) to allow only the authorised processes to read/write to them
- Files are quite slow relative to the mechanisms we are going to see later

In general, files are not used for IPC

But they should be considered as a choice when *huge* amounts of data need to be transferred

□

Inter-Process Communication

Files

- The file protections must be set properly (recall userids) to allow only the authorised processes to read/write to them
- Files are quite slow relative to the mechanisms we are going to see later

In general, files are not used for IPC

But they should be considered as a choice when *huge* amounts of data need to be transferred

Exercise. Read about the mechanism of choice to transfer the data describing the first ever image of a black hole (April 2019)

□