Processes

Daniel Bosk[1]

Department of Information and Communication Systems (ICS),
Mid Sweden University, Sundsvall.

process.tex 169 2016-11-09 06:52:00Z jimahl

Overview

1. Process Concept
   - What is a process?
   - In the OS
2. Process Scheduling
   - What is process scheduling?
   - Schedulers
3. Operations on Processes
   - How does processes come into being?
4. Interprocess Communication (IPC)
   - Independent vs cooperating
   - Different types of IPC
5. Testing the C examples in the book
   - Compiling

Literature

This lecture covers the first half of process management. It gives an overview of Chapter 3 “Process Concept” in [SGG13]

Overview

Mittuniversitetet
MID SWEDEN UNIVERSITY

What is a process?

- Process is the general term for CPU-activity.
- The basic executing entity in an operating system (OS).
- Process is a program in execution.
- Program is not a process:
    - Run the same program several times will spawn many unique processes.

What is a process?

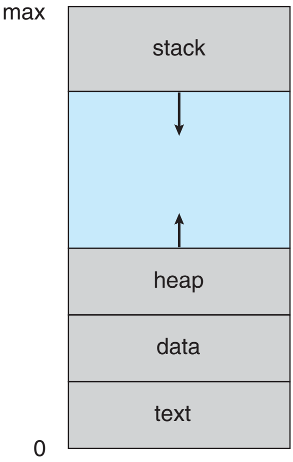

Figure: An schematic of an in-memory process. Image: [SGG09, Fig. 3.1, p. 102].

In the OS

- Must share resources fairly among all processes.
- OS has to keep track of all processes.
- OS uses a structure called Process Control Block (PCB) for this.

In the OS
PCB

- Process state
- Program counter
- CPU-registers
- Scheduling information
- Memory-information
- Accounting information
- I/O-information

In the OS

State What state the process is currently in:

- New,
- Running,
- Waiting,
- Ready, and
- Terminated.

Mittuniversitetet
MID SWEDEN UNIVERSITY

In the OS

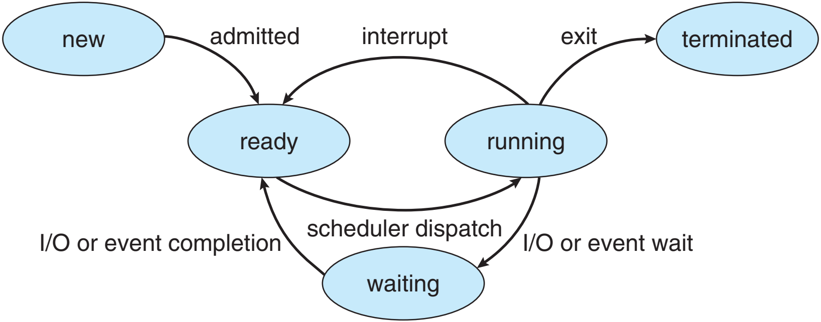

Figure: State-transition diagram. Image: [SGG09, p. 103].

In the OS

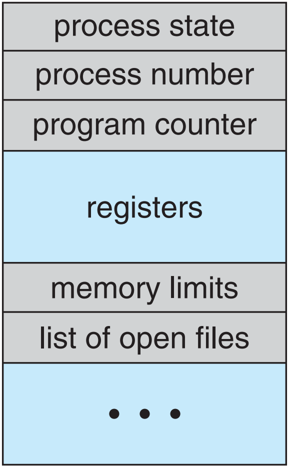

Figure: The PCB. Image: [SGG09].

In the OS

Program counter	Where in the program code the process is currently executing.
CPU-registers	A copy of the values in all CPU-registers, as the CPU-registers are used by the currently executing process.
Scheduling info	Priorities, scheduling queues.
Memory-info	Base and limit registers, page tables.
Accounting	How much CPU-time spent, how much I/O-time spent, time left of time-slice, process ID.
I/O-info	Allocated I/O-devices, open files.

Overview

Mittuniversitetet
MID SWEDEN UNIVERSITY

What is process scheduling?

- In a multiprogramming system, the main idea is to use the resources fully – never idly waiting around.
- When one process is waiting for I/O, let another process execute on the CPU.
- How to manage this? By using a scheduler.

What is process scheduling?

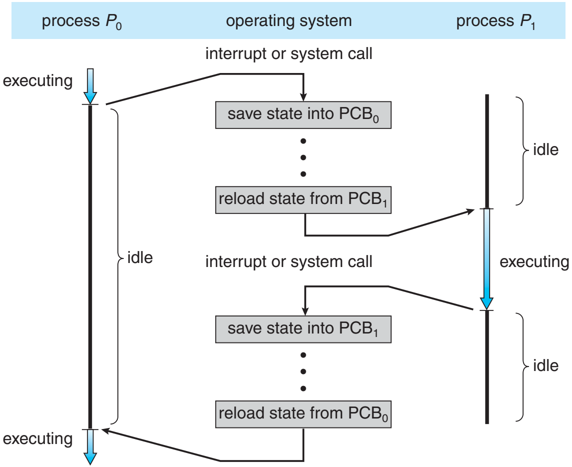

Figure: Context switch from one process to another. Image: [SGG09].

Schedulers

- There are three types of schedulers:
  - Long-term scheduler,
  - Mid-term scheduler, and
  - Short-term scheduler.
- These are used by different systems, and not necessarily in this structure and by these names.
- E.g. PC:s usually only have a short-term scheduler.

Schedulers

Long-term scheduler Comes from purely multiprogramming systems. Jobs are submitted to the system, they are stored on disk. Eventually the long-term scheduler selects a job from disk and loads it into memory.

Medium-term scheduler This scheduler is used for swapping out processes to disk, to free some more memory for the currently executing processes.

Short-term scheduler (Also CPU-scheduler) This scheduler chooses which process in memory is to be allowed to execute on the CPU. If one process blocks for I/O, then the short-term scheduler quickly chooses another process and allocates it to the CPU.

Schedulers

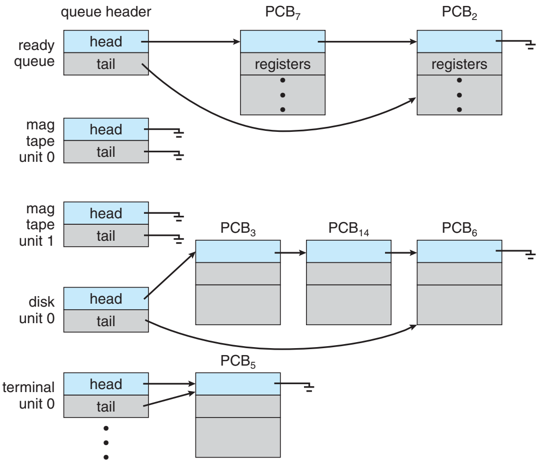

Figure: Different queues with some PCBs in them. Image: [SGG09].

Schedulers

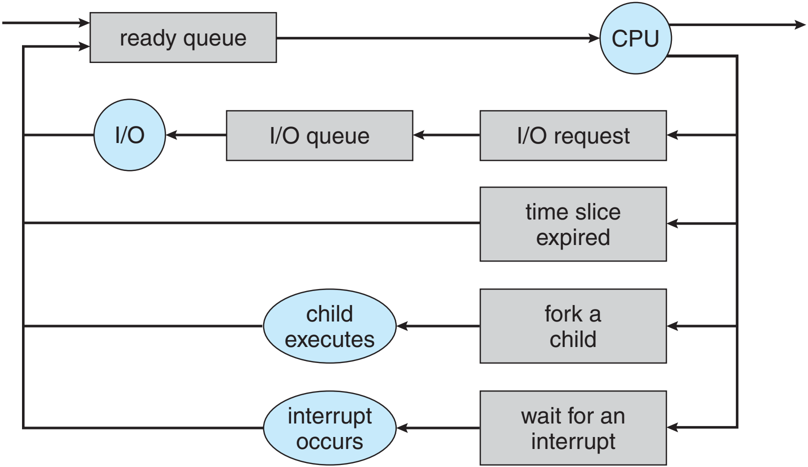

Figure: An illustration of a process's execution. Image: [SGG09].

Overview

1. Process Concept
   - What is a process?
   - In the OS
2. Process Scheduling
   - What is process scheduling?
   - Schedulers
3. Operations on Processes
   - How does processes come into being?
4. Interprocess Communication (IPC)
   - Independent vs cooperating
   - Different types of IPC
5. Testing the C examples in the book
   - Compiling

Mittuniversitetet
MID SWEDEN UNIVERSITY

How does processes come into being?

- Upon boot, the kernel is loaded.
- The kernel in turn starts the first process: init, with PID 1.
- A process can spawn other processes, i.e. children, and thus being their parent.
- This can be done with the fork(2) system call.

How does processes come into being?

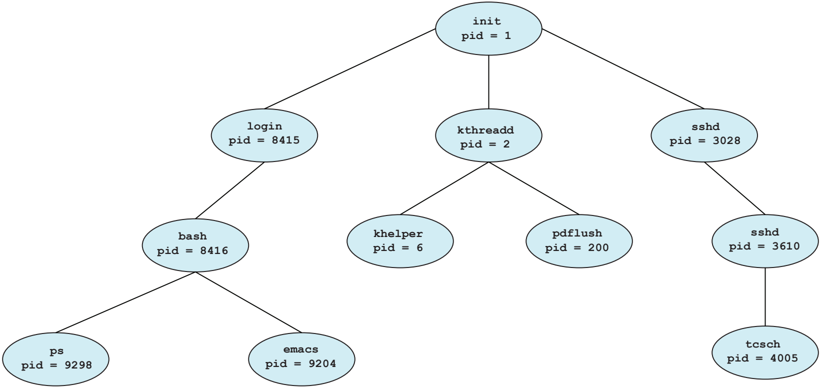

Figure: An example of a process tree from a Solaris system. Image: [SGG09].

How does processes come into being?

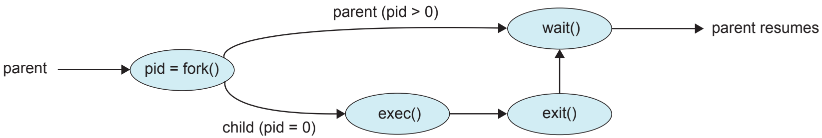

Figure: An illustration of the UNIX fork(2) system call. Image: [SGG09].

How does processes come into being?

- The parent then waits for its children to finish, using the wait(2) system call.
- An alternative is to kill the children, using the kill(2) system call.
- If the parent exits, using the exit(2) system call, then the children are assigned a new parent, namely the special init process.

How does processes come into being?

- We also have the exec(2) system call, which replaces the calling process with another process.

Overview

1. Process Concept
   - What is a process?
   - In the OS
2. Process Scheduling
   - What is process scheduling?
   - Schedulers
3. Operations on Processes
   - How does processes come into being?
4. Interprocess Communication (IPC)
   - Independent vs cooperating
   - Different types of IPC
5. Testing the C examples in the book
   - Compiling

Independent vs cooperating

- A process can be independent if it works on its own.
- If it doesn't work on its own, it's cooperating, i.e. it can affect other processes behaviour.

Independent vs cooperating

Reasons for cooperating

Information sharing Concurrent access to the same file by several users.

Computation speed-up Break a given task into subtasks and let different processes execute the different subtasks, the collect the results.

Modularity Replacing one process if we want to do one part differently.

Different types of IPC

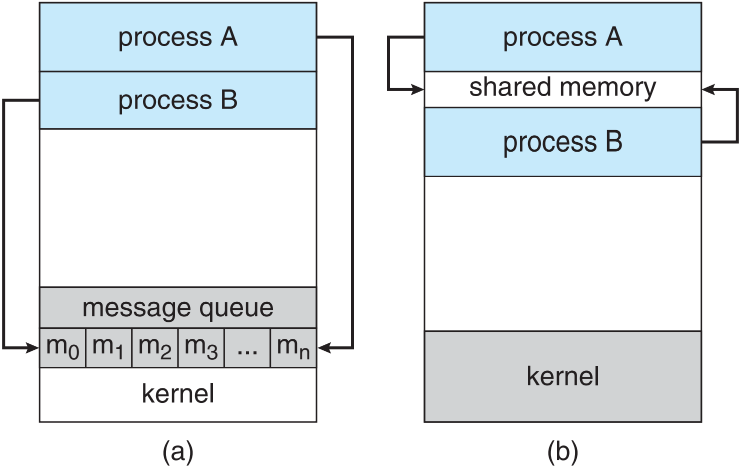

Figure: Two categories of IPC: (a) message-passing, (b) shared memory. Image: [SGG09].

Different types of IPC

- Several ways to implement message-passing.
- A few examples are:
    - Pipes,
    - Local procedure calls (LPC),
    - Sockets,
    - RPC (uses sockets),

Different types of IPC

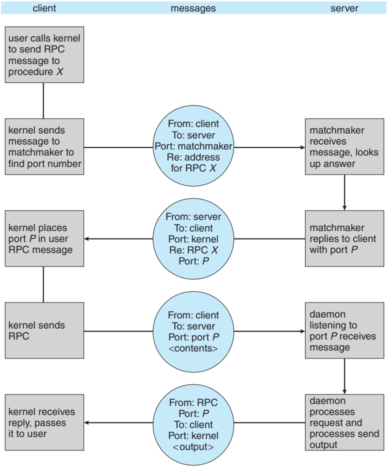

Figure: An example of an RPC. Image: [SGG09].

Mittuniversitetet
MID SWEDEN UNIVERSITY

Overview

1. Process Concept
   - What is a process?
   - In the OS
2. Process Scheduling
   - What is process scheduling?
   - Schedulers
3. Operations on Processes
   - How does processes come into being?
4. Interprocess Communication (IPC)
   - Independent vs cooperating
   - Different types of IPC
5. Testing the C examples in the book
   - Compiling

Compiling

- Download source from website, or just write it in any text editor (same as for Python).
- Go to the directory where the source file is stored.
- Compile the source code into executable code.
- Run!

Compiling

```
$ ${EDITOR} example.c
$ cc -o example example.c
[some output]
$ ./example
```

Mittuniversitetet
MID SWEDEN UNIVERSITY

Referenser I

Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. 8th ed. International Student Version. Hoboken, N.J.: John Wiley & Sons Inc, 2009.

Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. 9th ed. International Student Version. Hoboken, N.J.: John Wiley & Sons Inc, 2013.