Introduction

Johan Montelius

KTH

2020

What is an operating system?

What is an operating system?

Abstraction, virtualisation and managing of resource.

What is an operating system?

Abstraction, virtualisation and managing of resource.

What is an operating system?

Abstraction, virtualisation and managing of resource.

- Abstraction

What is an operating system?

Abstraction, virtualisation and managing of resource.

- Abstraction
  - How do we create an abstraction layer that provides an environment for programming of a process?

What is an operating system?

Abstraction, virtualisation and managing of resource.

- Abstraction
  - How do we create an abstraction layer that provides an environment for programming of a process?
- Virtualisation
  - How do we create the image of dedicated hardware while in fact we have several process sharing the same hardware?

What is an operating system?

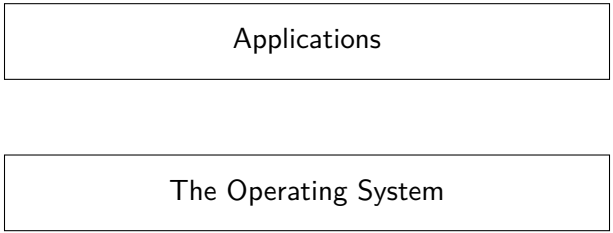
Abstraction, virtualisation and managing of resource.

- Abstraction
  - How do we create an abstraction layer that provides an environment for programming of a process?
- Virtualisation
  - How do we create the image of dedicated hardware while in fact we have several process sharing the same hardware?
- Resource management
  - Given that we have limited amount of resources, how do we share them in a fair way?

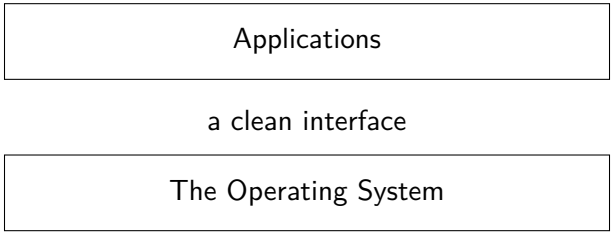
Abstraction

The Operating System

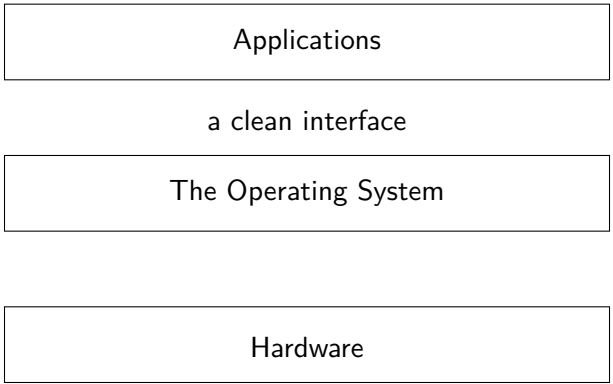
Abstraction



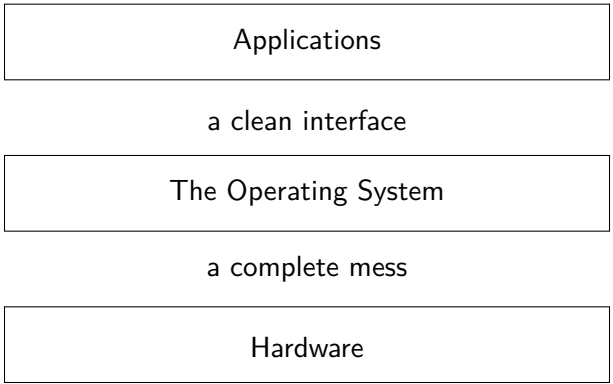
Abstraction



Abstraction



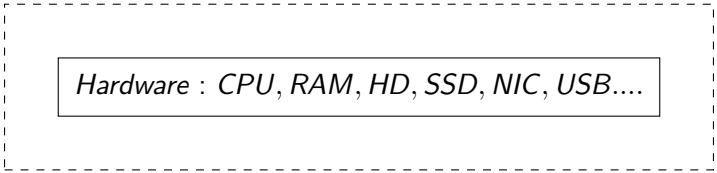
Abstraction



Abstraction

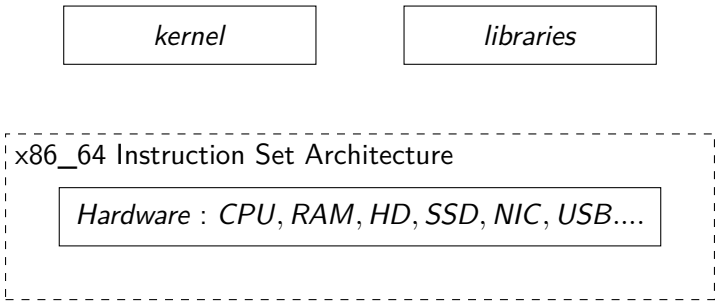
$Hardware : CPU, RAM, HD, SSD, NIC, USB....$

Abstraction

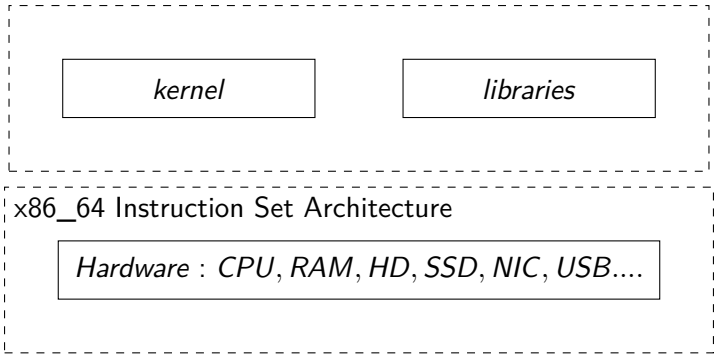


Abstraction

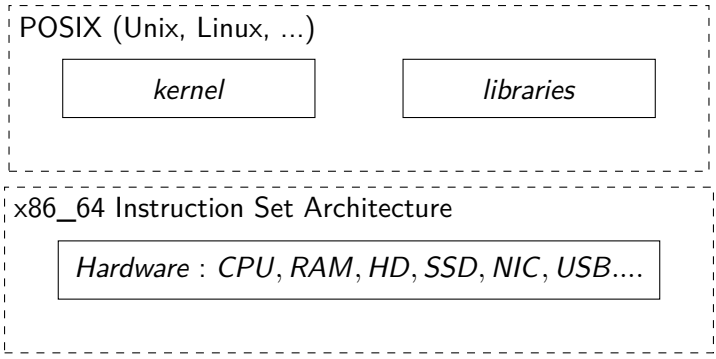
Abstraction



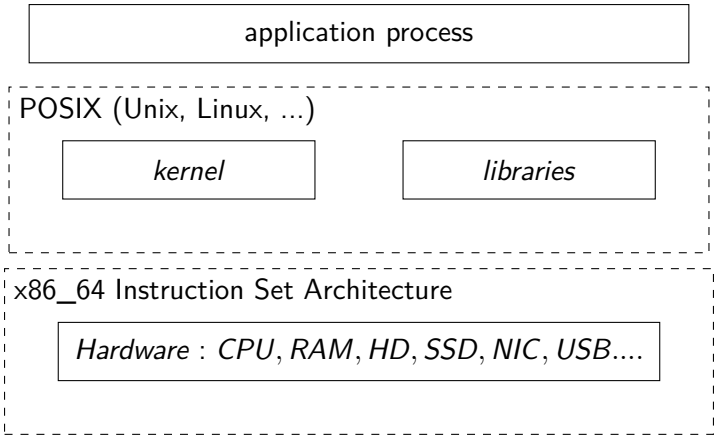
Abstraction



Abstraction



Abstraction



POSIX: Portale Operating System Interface

Operating system API

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..
- file operations: fopen, fclose, fread, fwrite, ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..
- file operations: fopen, fclose, fread, fwrite, ...
- ...

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..
- file operations: fopen, fclose, fread, fwrite, ...
- ...

Command Line Interpreter

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..
- file operations: fopen, fclose, fread, fwrite, ...
- ...

Command Line Interpreter

- shell: the text based interface

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..
- file operations: fopen, fclose, fread, fwrite, ...
- ...

Command Line Interpreter

- shell: the text based interface
- scripting languages

POSIX: Portale Operating System Interface

Operating system API

- process handling: fork, exec, wait, ...
- process communication: pipes, ..
- threads handling: pthread_create, ...
- managing directory and file ownership
- network handling: socket, listen, accept, ...
- ...

The C Standard Library (ISO C18)

- memory allocation: malloc, free, ...
- signal handling: signal, raise, kill, ..
- file operations: fopen, fclose, fread, fwrite, ...
- ...

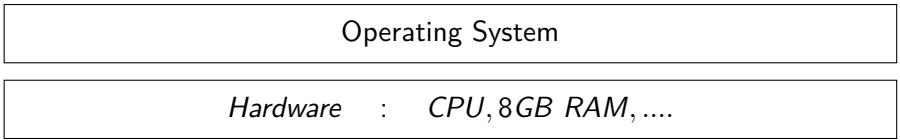
Command Line Interpreter

- shell: the text based interface
- scripting languages
- ...

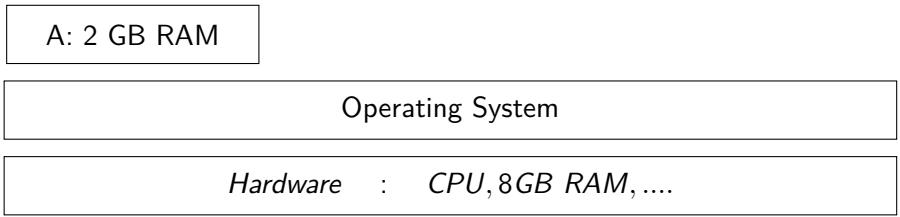
C programs

```
int counter = 0;

void hello(char *name){
  printf("Hello: %s, %d\n",  name, counter);
}

int main() {
  char *me = argv[1];
  while(counter != 10) {
     counter++;
     hello(me);
     sleep(1);
   }
   return 0;
 }
```

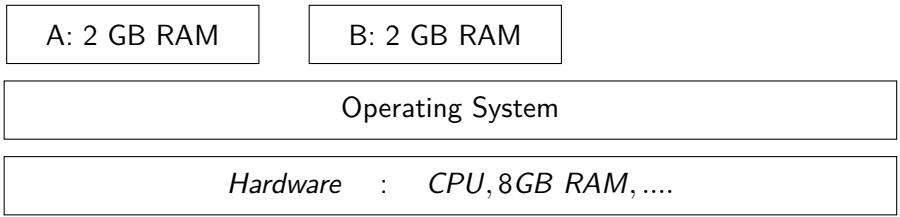
Virtualization



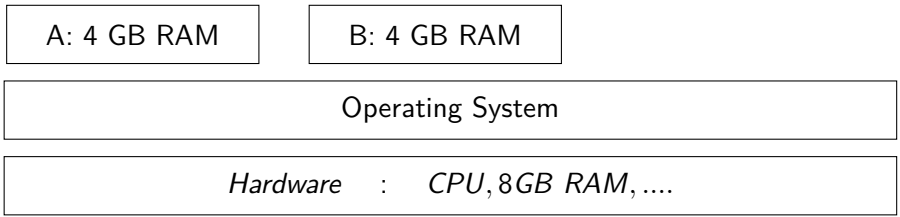
Virtualization



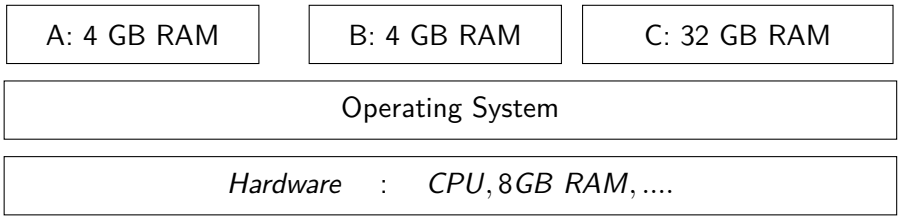
Virtualization



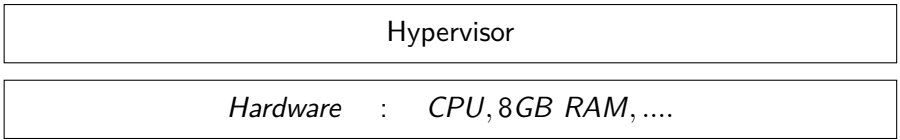
Virtualization



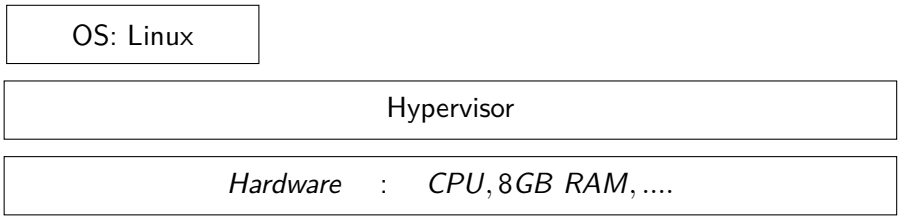
Virtualization



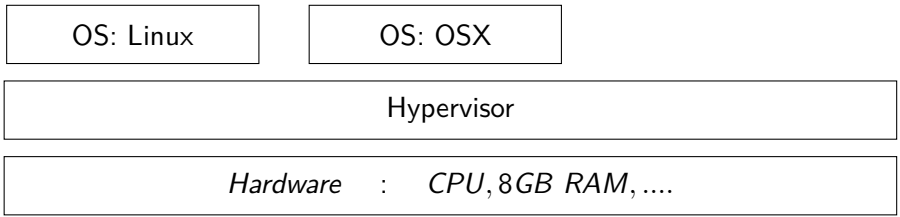
Virtualization



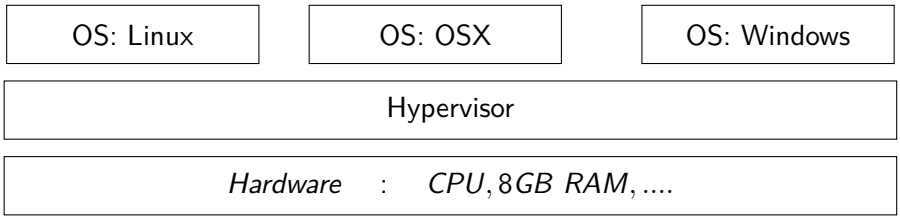
Virtualization



Virtualization



Virtualization



Resource management

Resource management

- Time: scheduling, how do we divide the execution time among processes

Resource management

- Time: scheduling, how do we divide the execution time among processes
- Memory: efficient allocation and deallocation, malloc/free...

Resource management

- Time: scheduling, how do we divide the execution time among processes
- Memory: efficient allocation and deallocation, malloc/free...

to implement an operating system

Why is it hard to implement an operating system?

Summary

Start programming today.