Introduction

Johan Montelius

KTH

2019
Choosing an operating system
Choosing an operating system

What is important when choosing an operating system?

What everyone else is using.

Look and feel, ease of use.

Hardware

Cost

Availability of programs

How it works under the hood.
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

A well structured world

a clean interface

The Operating System

a complete mess

Hardware
Abstraction

application process

POSIX (Unix, Linux, ...)

kernel
libraries

x86_64 architecture

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

...
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

Operating System

<table>
<thead>
<tr>
<th>Hardware</th>
<th>CPU, 8GB RAM, ....</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: 4 GB RAM</td>
<td>B: 4 GB RAM</td>
</tr>
</tbody>
</table>
Virtualization

<table>
<thead>
<tr>
<th>OS: Linux</th>
<th>OS: OSX</th>
<th>OS: Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hypervisor</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td><strong>CPU, 8GB RAM,</strong></td>
<td></td>
</tr>
</tbody>
</table>


Resource management

- Time: scheduling, how do we divide the execution time among processes
- Memory: efficient allocation and deallocation, malloc/free...
- Storage: HDD, SSD, ....
Why is it hard to implement an operating system?
Start programming today.