



**FREE eBook**

# LEARNING operating-system

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#operating-  
system

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with operating-system.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
install the popular Clonezilla to clone a Mac OS hard drive (basic usage).....	2
<b>Chapter 2: User Space and Kernel Space.....</b>	<b>4</b>
Introduction.....	4
Examples.....	4
Switching between user space and kernel space.....	4
Exam CPU time allocation between user space and kernel space.....	4
Exam CPU time spent on one process.....	4
<b>Credits.....</b>	<b>6</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [operating-system](#)

It is an unofficial and free operating-system ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official operating-system.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with operating-system

## Remarks

This section provides an overview of what operating-system is, and why a developer might want to use it.

It should also mention any large subjects within operating-system, and link out to the related topics. Since the Documentation for operating-system is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

Detailed instructions on getting operating-system set up or installed.

### install the popular Clonezilla to clone a Mac OS hard drive (basic usage)

Naturally, being able to install hard drive cloning utilities can be an important aspect of installing and maintaining your operating system.

Getting set up with Clonezilla is surprisingly less straightforward than I'd have expected. The wealth of options, while valuable, also makes each part of identifying and loading the software overly convoluted. However, once it is set up, it's very easy to use.

Steps to download for Mac from <http://clonezilla.org/downloads/download.php?branch=stable> and make the following selections:

1. Select CPU architecture: `amd64`
2. Select file type: `iso`
3. Select repository: `auto`

Steps to format a bootable usb thumb drive

1. Open Applications > Utilities > Disk Utility
2. Insert thumb drive at least 1.5 GB in size
3. Provide a disk name and select format options `OS X Extended (Journaled)` and `GUID Partition Map`

Steps to install Clonezilla on the flash drive.

```
$ cd ~/
$ mv ~/Download/<clonezilla.iso> ~/ # being in the $HOME dir may or may not have mattered
$ hdiutil convert -format UDRW -o ~/clonezilla.dmg ~/<clonezilla.iso>
```

```
$ diskutil list # node device node assigned to flash media (e.g. if /dev/disk2, diskN is
disk2)
$ diskutil unmountDisk /dev/<diskN>
$ sudo dd if=/path/to/downloaded.img of=/dev/diskN # didn't succeed with rdiskN or with bs=1m
as other instructions stipulate
$ diskutil eject /dev/diskN
```

Reboot and hold alt key, then follow the on-screen instructions.

Useful references (though keep in mind that many of the details did not work out, where the above did):

- [http://drbl.org/faq/fine-print.php?path=./2\\_System/121\\_pxe\\_boot\\_mac\\_machine.faq#121\\_pxe\\_boot\\_mac\\_machine.faq](http://drbl.org/faq/fine-print.php?path=./2_System/121_pxe_boot_mac_machine.faq#121_pxe_boot_mac_machine.faq)
- <http://osxdaily.com/2015/06/05/copy-iso-to-usb-drive-mac-os-x-command/#comment-2000664>

Worth noting that once backup is complete, on-screen instructions continue to be useful. It's also easy to overlook the instruction for shutting down: `$ sudo shutdown -h now`

Read [Getting started with operating-system online](https://riptutorial.com/operating-system/topic/7472/getting-started-with-operating-system): <https://riptutorial.com/operating-system/topic/7472/getting-started-with-operating-system>

---

# Chapter 2: User Space and Kernel Space

## Introduction

The RAM can be divided into two parts. The kernel space and user space. The kernel runs in the kernel space, which no other programs can access. User programs have to run in user space. User space is a form of sand-boxing, where user programs can only access to memory that allocated to them so that they can't mess up other programs and the kernel. To use the system resource, a program uses system calls to access a certain part of the kernel space and back to user space when the call return.

## Examples

### Switching between user space and kernel space

Kernel manages operating system resources. User program can only access to those resources by making system calls to the kernel. System call is similar to an API of kernel, which in term, runs kernel tasks your program needs.

```
str = "something" // run on user space
x = x + 1 // run on user space
file.write(str) // switch to kernel space
y = x + 4 // switch back to user space
```

### Exam CPU time allocation between user space and kernel space

Use `top` command to exam CPU time allocation between user space and kernel space.

```
%Cpu(s): 24.8 us, 0.5 sy, 0.0 ni, 73.6 id, 0.4 wa, 0.0 hi, 0.2 si, 0.0 st
```

Explanation:

1. 24.8 us (user space): 24.8% of CPU time is spent on user process.
2. 0.5 sy (system): 0.5% of CPU time is spent on kernel space.
3. ni (niceness): the ratio of CPU time spent on low priority processes.
4. id (idle): the ratio of CPU time spent on idle processes (during the period of time, CPU can not do anything).
5. wa (wait): the ratio of CPU time spent on waiting for I/O (during the period of time, CPU can not do anything).
6. hi (hardware interrupt): the ratio of CPU time spent on responding hardware interruption.
7. si (software interrupt): the ratio of CPU time spent on responding software interruption.
8. st (stole time): the ratio of CPU time stolen by virtual machine.

### Exam CPU time spent on one process

## Use time command.

```
time ./perl-timeout-example 100.100.100

We could not ping the desired address!

real    0m5.0013s
user    0m0.004s
sys     0m0.008s
```

- **Real:** Total time from start to finish of the call, including the CPU time spent on other processes.
- **User:** The amount of CPU time spent in user space.
- **Sys:** The amount of CPU time spent in the kernel space.

Normally, user + sys is the total CPU time spent on such process. It is smaller than real. But in multicore CPU, user + sys is the total CPU time spent on this process. It could be large than real.

Read User Space and Kernel Space online: <https://riptutorial.com/operating-system/topic/9795/user-space-and-kernel-space>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with operating-system	<a href="#">Community</a> , <a href="#">Downvoter</a> , <a href="#">ownsourcing dev training</a>
2	User Space and Kernel Space	<a href="#">Junbang Huang</a>