

Introduction

An introduction to what processes are and what they are comprised of. Followed by a more in-depth examination of threads, their concept, usefulness and finally how you create your own using a POSIX compliant API as that which is available in Linux.

Contents

Theme(s)	Topic(s)	Reference
Parallel programs	A case study	[1]
Processes	Virtual Memory	[1, 2, 4] [3, Chap. 4, p. 131-141]
	Anatomy of process	[1, 4]
Threads	Models	[1, 2, chap. 5]
	Posix threads (using 'em)	[6, chap. 29]
	Why threads	[5]

Material

Slides

- [1] S. Hansen, *Parallel programs, processes and threads*, Slides - see course repos.

Local repository

- [2] R. B. Muhammed, *Introduction to operating systems*, TFJ composed a pdf based on the text from: <http://www.personal.kent.edu/%7ermuhamma/OpSystems/os.html>.
- [3] S. E. David Mosberger, *IA-64 Linux Kernel: Design and Implementation*. Prentice Hall, 2002, Only one chapter - Chapter 4 Virtual Memory, filename: Virtual Memory.pdf, ISBN: 978-0-13-061014-0.

Online

- [4] G. Duarte. (). Anatomy of a program in memory, [Online]. Available: <http://duartes.org/gustavo/blog/post/anatomy-of-a-program-in-memory/>.
- [5] H. Sutter. (). The free lunch is over, [Online]. Available: <http://www.gotw.ca/publications/concurrency-ddj.htm>.

Hardback

- [6] M. Kerrisk, *The Linux Programming Interface*. No Starch Press, Inc, 2010, ISBN: 978-1-59327-220-3.

Take special of note of the reference [3, Chap. 4], since the title does not refer to the actual filename, however 'Virtual Memory.pdf' does as denoted in the comment.

Fundamental questions to consider while reading

Processes and threads Course specifics

- Processes
 - What is a virtual memory
 - What purpose does the MMU serve
 - What is a scheduler
 - What is the context concept and what happens when a *context switch* occurs
 - What are the different sections which a processes can be dissected into - its anatomy
 - Which states can it be in - birth, life & death (more to it than that)
- Threads
 - What is a thread
 - Why do you want to use threads
 - What kind of trouble do they present
 - Threading model
 - * Which models exists
 - * Their properties
- Using threads in Linux
 - What is a *function pointer* or *call back function*
 - How do you create a thread
 - How do you terminate a thread