



# OPERATING SYSTEMS

**Certificate awarded by:** ECE Paris

**Welcome event:** July 3<sup>rd</sup>, 2017 (morning)    **Start date of courses:** July 3<sup>rd</sup>, 2017 (afternoon)

**End date:** July 27<sup>th</sup>, 2017    **Certificate Ceremony:** July 27<sup>th</sup>, 2017

**Total ECTS:** 8    **Total contact hours:** 78

**Program requirement:** a minimum 18 years of age

**Program location:** ECE Paris - Campus Eiffel I, 10 Rue Sextius Michel, 75015 Paris, France

**Language of instruction:** English

**PROGRAM FEE: 1,850€**

## FEE INCLUDES:

- Orientation/Welcome Event
- Weekly cultural visits/activities
- Computer accounts at the school (WIFI access)
- Access to the school's MediaCenter
- Official transcript of grades
- Program Certificate
- Certificate Ceremony

## COURSES

Course Title	ECTS (credits)	Contact hours	Level (undergraduate or graduate)
Advanced Programming	4	39	undergraduate
Operating Systems	4	39	undergraduate

**CONTACT** [summer@pariseiffel.fr](mailto:summer@pariseiffel.fr)

# PROGRAM COURSE DESCRIPTIONS

## COURSE 1

### COURSE OVERVIEW:

This is an intermediate level computer science course in which programming concepts and techniques will be studied in order to construct computer applications, which have an interface to the operating system. This class requires previous knowledge in programming, TOS structures, and operating systems. The learning outcome for this course is for students to develop programs in which the concepts of concurrent and multi-core programming are put into practice.

Upon completion of this course, students will have acquired an advanced knowledge of the development of programs in

C, their debugging and implementation for the design and development of computer applications that optimize the utilization of the operating system nucleus resources; fully understand operating system process management and the synchronization and communication techniques between the processes, as well as the advantages of developing concurrent and multithreaded algorithms simultaneously in order to implement them using tools that guarantee their efficiency.

Course Title	Advanced Programming
Learning outcomes	Learn how to use the different function/system calls in order to : <ol style="list-style-type: none"><li>1. manage processes</li><li>2. manage the memory</li><li>3. manage files</li><li>4. manage I/O</li></ol>
Pre-requisites	C programming

### COURSE OUTLINE:

- User's View
  - linux commands
  - scripting
- Programmer's View
  - Creating processes and running applications
  - Implement « system » and program a simple shell (managing files)
  - Implement « ls » (file meta-information)
  - Parallelizing calculations using processes and shared memory
  - I/O management – using sockets to create a client and a server (distant calculator with multiple clients, chat application)
  - I/O management – implementing redirections
  - I/O management – using unnamed and named pipes

### COURSE 2

#### COURSE OVERVIEW:

Intermediate computer course that provides students with knowledge of the functioning and interaction of the components of operating systems. Students will use a programming language that is compatible with open-source OS in order to apply the basic concepts of operating systems. This will allow students to understand an open-source operating system and give them the bases for managing and evaluating operating systems, as well as the opportunity to apply the operating-system concepts and techniques in other environments. Students will complete programming exercises related to these topics. This course requires prior knowledge

of programming and data structures. Learning outcome: students will solve problems related to the basic algorithms used in diverse components of an operating system. They will demonstrate their competent handling of commands, system calls and device management for specific Unix-type operating systems in the laboratory. Programming using calls to systems such as fork, system and basic concurrent programming for managing processes and threads (POSIX). Development of programs to add functionality to the Operative System nucleus applying the concepts learned in the classroom.

Course Title	Operating Systems
Learning outcomes	Understand how the operating system manages its resources (processor, memory, Filesystems, I/O)
Pre-requisites	C programming

#### COURSE OUTLINE:

- Process Management
- Threads vs Processes
- Synchronization (Theory and Practice)
- Memory Management
- Paging implementation
- Filesystems (FAT, Inode, ...)
- I/O Management