

Adrien Forest
fjxokt@gmail.com
Christophe Carasco
carasco.christophe@gmail.com
Matthieu Maury
mayeu.tik@gmail.com



UPPSALA
UNIVERSITET

*Department of Information Technology
Master Student*

Operating System Project 10hp

SSIK : Simply & Stupidly Implemented Kernel Architecture Sketch

April 13, 2010

Contents

1	Modules	3
1.1	General	3
1.2	Description	3
1.2.1	Process	3
1.2.2	Scheduler	3
1.2.3	Error	3
1.2.4	Message Handling	3
1.2.5	System Library	3
1.2.6	Kernel Library	4
1.2.7	User software	4
1.2.8	Kernel	4
1.2.9	Memory management (optional)	4
2	Flow	5
2.1	System init flow	5
2.2	Process creation	5
2.3	Message sending	5

1. Modules

1.1 General

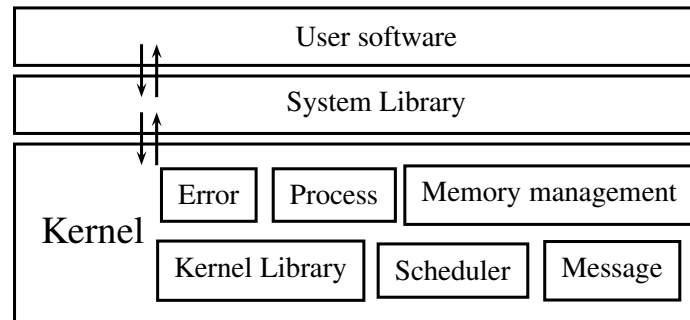


Figure 1.1: General Kernel overview

1.2 Description

1.2.1 Process

The process module handles process creation, modification, information reading, settings supervised and supervision process etc. Also the management of lists of process (push/pop in the list, search in the lists, sorting, etc.)

1.2.2 Scheduler

The scheduler will be in charge of choosing and picking the next process to run by following the rule of higher priority first and round robin for egal priority process. This module will use the process module to search and read information about the processes to be able to choose the next process.

1.2.3 Error

This module will provide some facilities to print diagnostics and handle errors (like the perror())

1.2.4 Message Handling

This module will provide function to pass message to different processes. It also provides functions to create/send/read/destroy messages and also manage a list of message.

1.2.5 System Library

The system library will provide some basic and useful functions (e.g. printf(), getc()) through system calls to the kernel. This will also provide an interface to the other modules of the kernel for the user.

This function will only be used by user softwares, kernel software using the kernel library.

1.2.6 Kernel Library

The kernel library will provide, at least, the same basic and useful functions as the system library but "directly" (i.e. without syscalls). This function will only be called by kernel software.

1.2.7 User software

This module will provide a bunch of user program such as a shell, current process information, system information, requirement user program, etc.

1.2.8 Kernel

The "kernel module" will be in charge of setting the basic needs of the kernel (default value, list of process, etc.) and launch the init process.

1.2.9 Memory management (optional)

This optional module will provide some dynamic memory management functions. Without this module, all the lists (such as process, message, etc.) will have a static size.

2. Flow

2.1 System init flow

- print some information (OS name, developer names, system version, etc.)
- allocate and initialize all the needed structures for process management
- spawn the init process
- finish the initialization inside the init process
- init spawn the text scroller process
- make init the supervisor of the text scroller process
- init print that the system is ready
- init spawn a shell

2.2 Process creation

- make a system call (if you are in the user space)
- pass the name of the program
- check if the program exists
- look for available space for the process
- allocate the space (if dynamic memory management is set)
- initialize the PCB with default values and the program
- place the process in the ready queue
- return to the caller

2.3 Message sending

- make a system call (if you are in the user space)
- pass the message and the process ID to the syscall
- check if the process ID is valid
- add the message to the messages list of the process if there is space in the message list.
- return to the caller