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

University year 2009-2010

Written with LaTeX

Contents

1. Modules

1.1 General

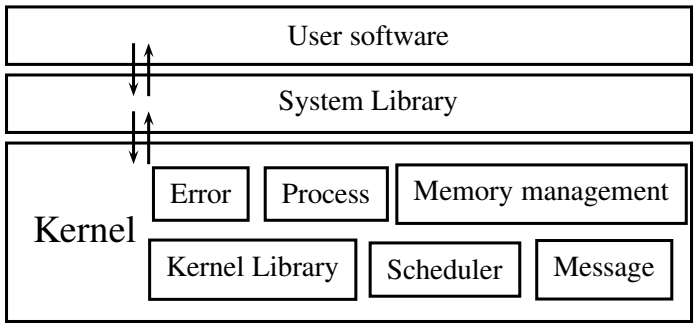

Figure 1.1: General Kernel overview

1.2 Description

1.2.1 Process

The process module handle 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 to choose and pick 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, read information about the process to be able to choose the next process.

1.2.3 Error

This module will provide some facility to print diagnostic and handle error (like the perror())

1.2.4 Message Handling

This module will provide function to pass message to different process. It also provie function to create/send/read/destroy message and also managing a list of message.

1.2.5 System Library

The system library will provide some basic and usefull function as printf(), getc(), etc. through system call to the kernel. This will also provide an interface to the other module of the kernel for the user.

This function will only be use by user software, kernel software will use the kernel library.

1.2.6 Kernel Library

The kernel Library will provide, at least, the same basic and usefull functions as the system library but “directly” without syscall. This function will only be called by kernel software.

1.2.7 User software

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

1.2.8 Kernel

The “kernel module” will be in charge to setup the basic need of the kernel (default value, list of process, etc.) and launch the init process.

1.2.9 Memory management (optionnal)

This optionnal module will provide some dynamic memory management function. Without this module all the list (sure as process, message, etc.) will hava a static size.

2. Flow

2.1 System init flow

- print some informations (OS name, develloper 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 exist
- look for availlable space for the process
- allocate the space (if dynamic management is set)
- initialize the PCB with default value and the program
- placing 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