# *RongOS* — A simple operating system implementation

Start from scratch

Qiyuan Pu

School of Big Data and Intelligence Engineering
Southwest Forestry University

RongOS Introduction, April 2018

## Table of Contents

## Table of Contents

## Operating System Introduction

An operating system (OS) is system software that manages computer hardware and software resources and provides common services for computer programs.

**Service**

1. Context Switching and Scheduling.
2. Memory Management.
3. Interprocess Communication.
4. File Systems.
5. High level I/O facilities.

## Table of Contents

## The Overall Design of *RongOS*

RongOS uses a layered and modular design. Includes the following three layers.

- User Layer
- Kernel Layer
- Hardware Control Layer

RongOS has the following modules.

- API.
- Scheduler.
- Memory Management.
- FAT.
- Driver.
- Graphic.
- System Call.

## The Overall Design of *RongOS*

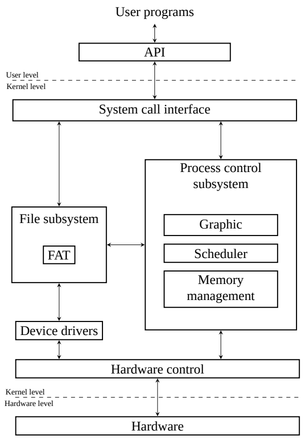


Figure: *RongOS* overall design diagram.

## Table of Contents

## Process Management

Process management is the most complex part of operating system development, but this part has far-reaching and extensive impact on the performance of the entire system.
Process management needs to complete the following work.

- Add task to task pool.
- Scheduling running.
- Sleeping task.
- Process communication.

## Process Management

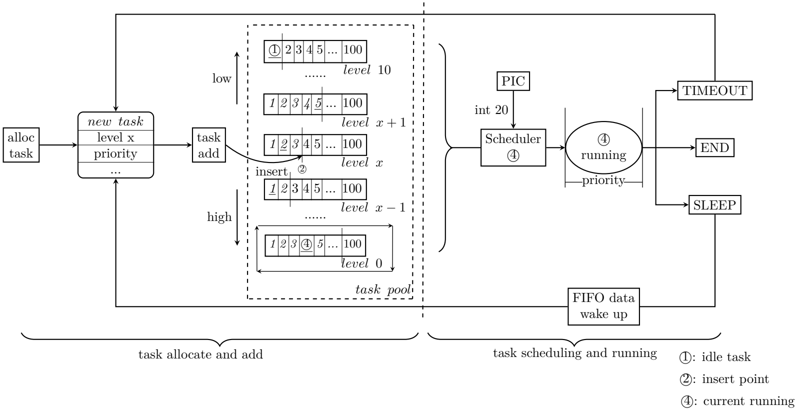


Figure: Process management overall design diagram.

## Memory Management

Memory management is the process of controlling and coordinating computer memory, assigning portions called blocks to various running programs to optimize overall system performance.
Memory management needs to complete the following work.

- Memory allocation.
- Memory release.

The memory release is more complicated. It is divided into three situations.

- Can be merged with the previous one.
- Can be merged with the latter
- It can neither be merged with the latter one nor the former one.

## API

The API is the interface that provides applications access to the operating system kernel. More than twenty APIs are provided in *RongOS*. Examples are as follows.

**Examples**

- void api_putchar(int c): output a character c on the console window.
- void api_putstr0(char *s): ouput a string s on the console window.
- int api_getkey(int mode): accept keyboard input, mode indicates whether the keyboard is sleeping.
- ......

## FAT

The FAT records which blocks the files are stored in. The file system of this system is relatively simple and only supports FAT.
The main functions are as follows.

- file_readfat: read FAT table.
- file_loadfile: load file.
- file_search: search file.

## Driver

In computing, a device driver is a computer program that operates or controls a particular type of device that is attached to a computer. A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without needing to know precise details of the hardware being used.
There are several device drivers in *RongOS*.

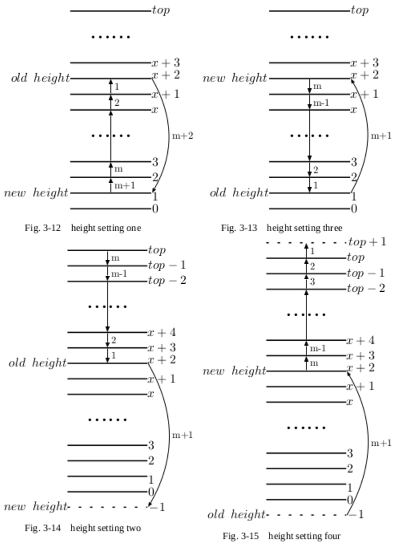
- Keyboard.
- Mouse.
- Palette.

## Graphic Management

Usually graphics management is not the concern of the operating system kernel for Linux, but not for Windows. However, as a simple system with windows, it is necessary to give some instructions for graphic management. The main services provided by graphic management are as follows.

- Refresh the window.
- Set the height of window.

# Set Height of Window



top
x + 3
x + 2
x + 1
x
old height
new height
m+2
m+1
m
m-1
top + 1
top − 1
top − 2
x + 4
3
2
1
0
−1
Fig. 3-12 height setting one
Fig. 3-13 height setting three
Fig. 3-14 height setting two
Fig. 3-15 height setting four

Figure: Set the height of window.

## Table of Contents

## Stars2

Obviously, application development is not the content of operating system development. But in order to verify that *RongOS* is able to run programs developed in accordance with the *RongOS* API, I decided to introduce an application implementation that is stars2. This application is to imitate the stars in the sky, I think the stars are a symbol of dreams, and I intend to end my thesis by showing dream. And developing an operating system is one of my dreams. The project ends but the dream will not.

# Trap

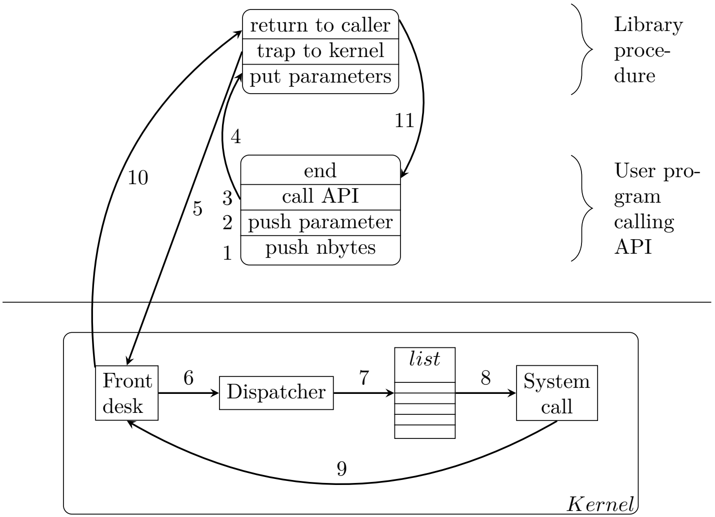


Figure: A trap when a APP call API provided by *RongOS*.

## Result of Stars2

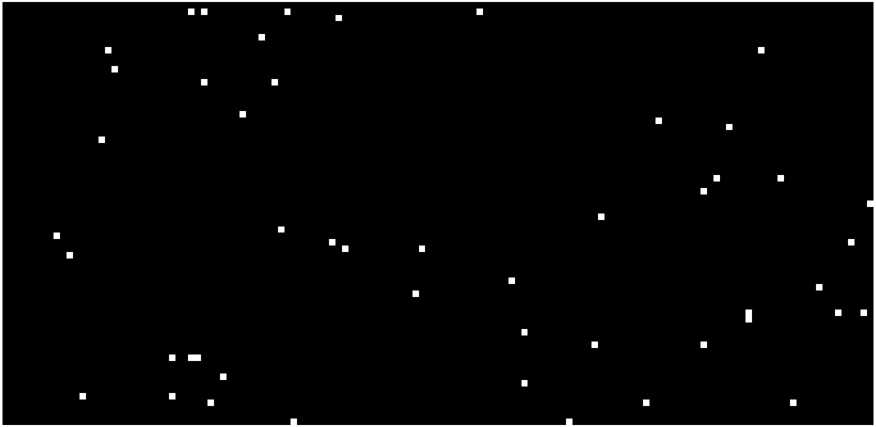
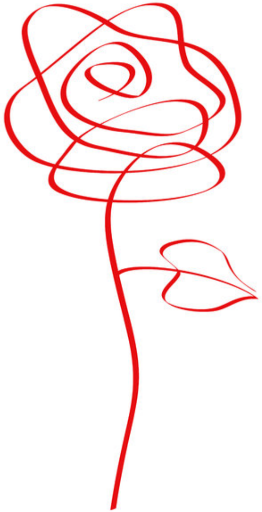


Figure: The running result of stars2.

## Table of Contents

## Postscript — Why Rose

*Why*  *mascot?*

## Postscript — The Short Answer

$$\begin{tikzcd}\boxed{RongOS} \arrow[r] & \boxed{ROS} \arrow[r] & \boxed{rose}\end{tikzcd}$$

*But why RongOS?*

Eh...that's another new and long story...

## Postscript — Thanks

Thank you for your interest in RongOS!