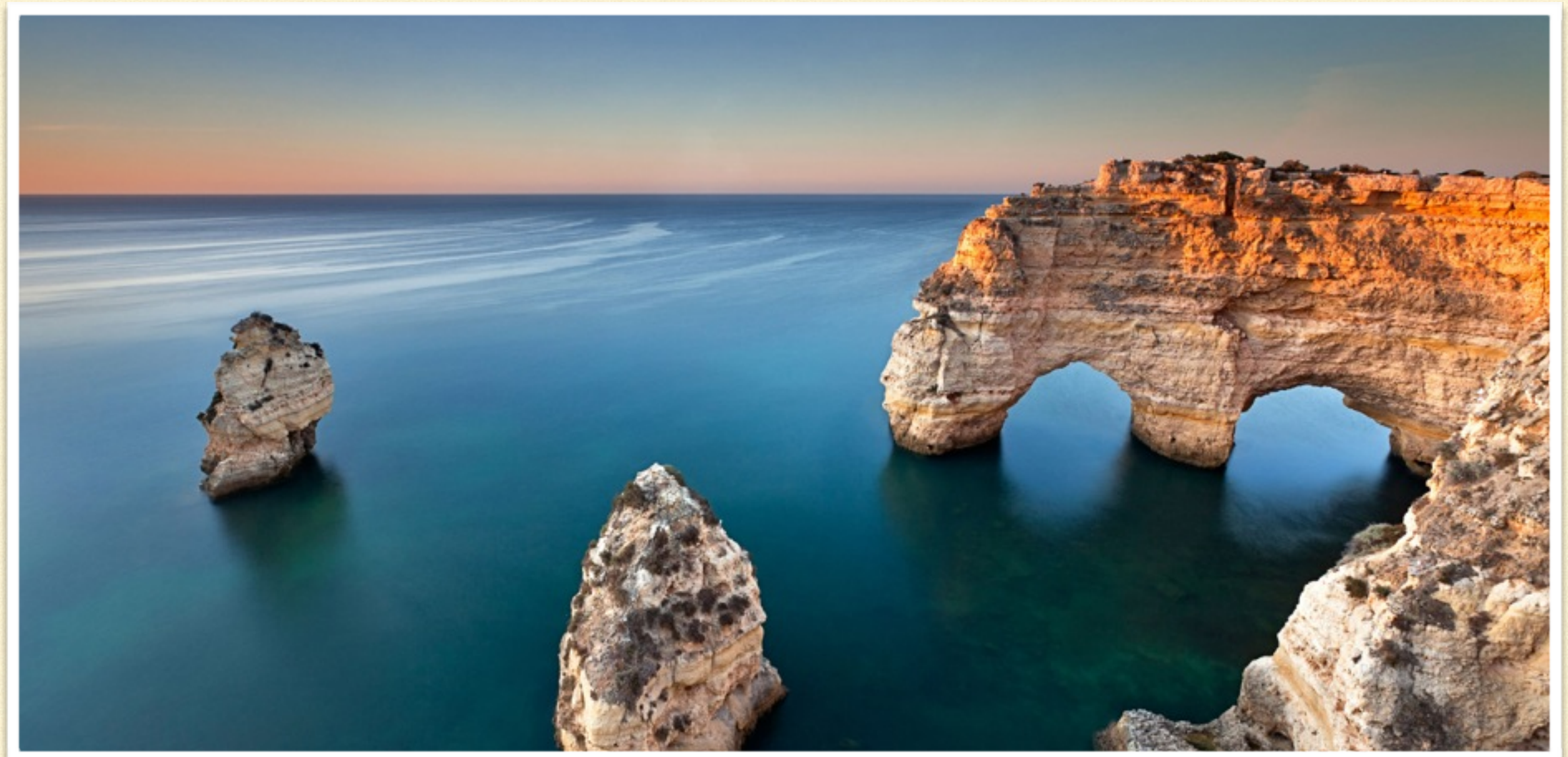
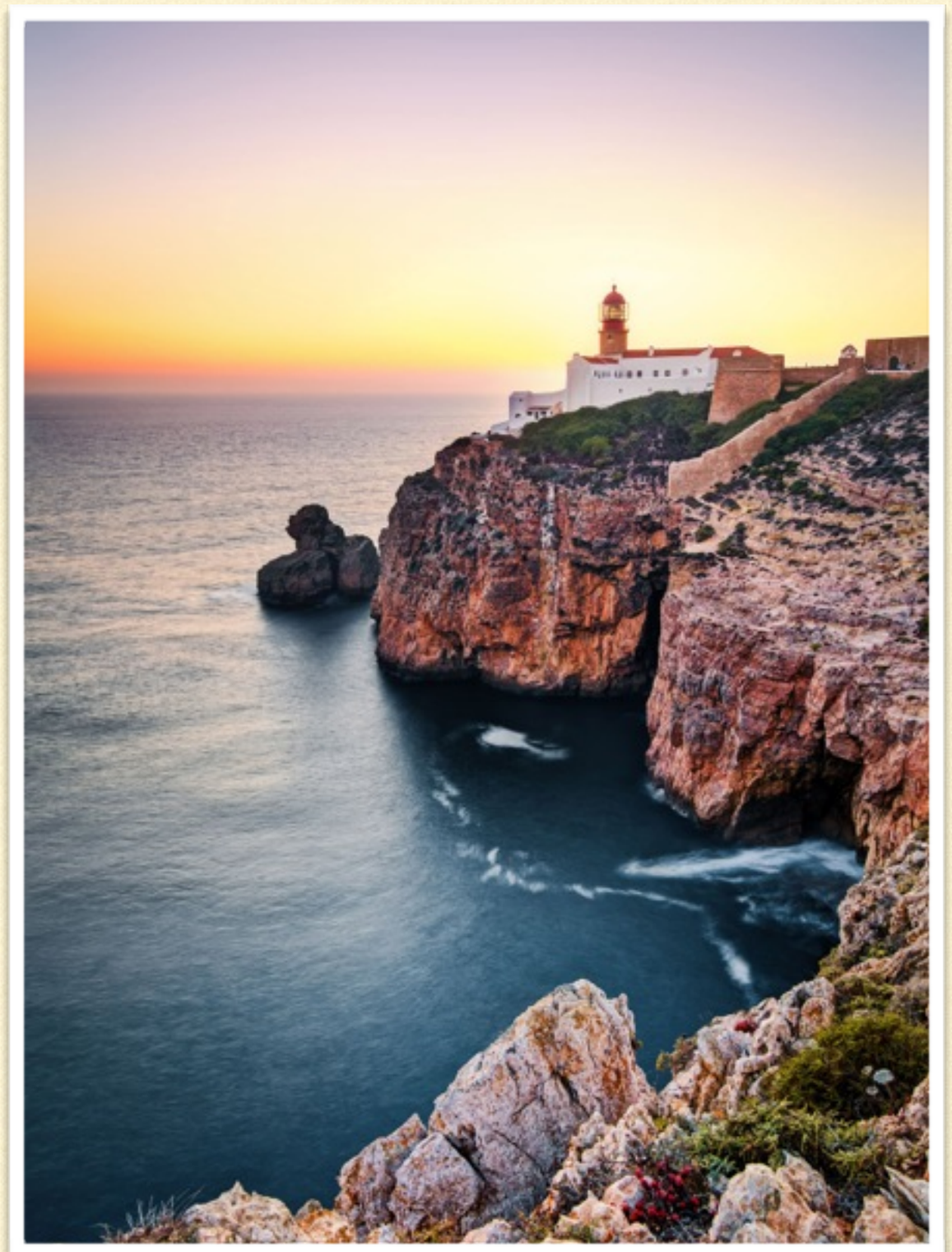


信号量在LINUX和JOS上的实现

陈诗安

大纲

- 简介
- Linux(3.12.1)中信号量的实现
- JOS中信号量的实现
- Tests



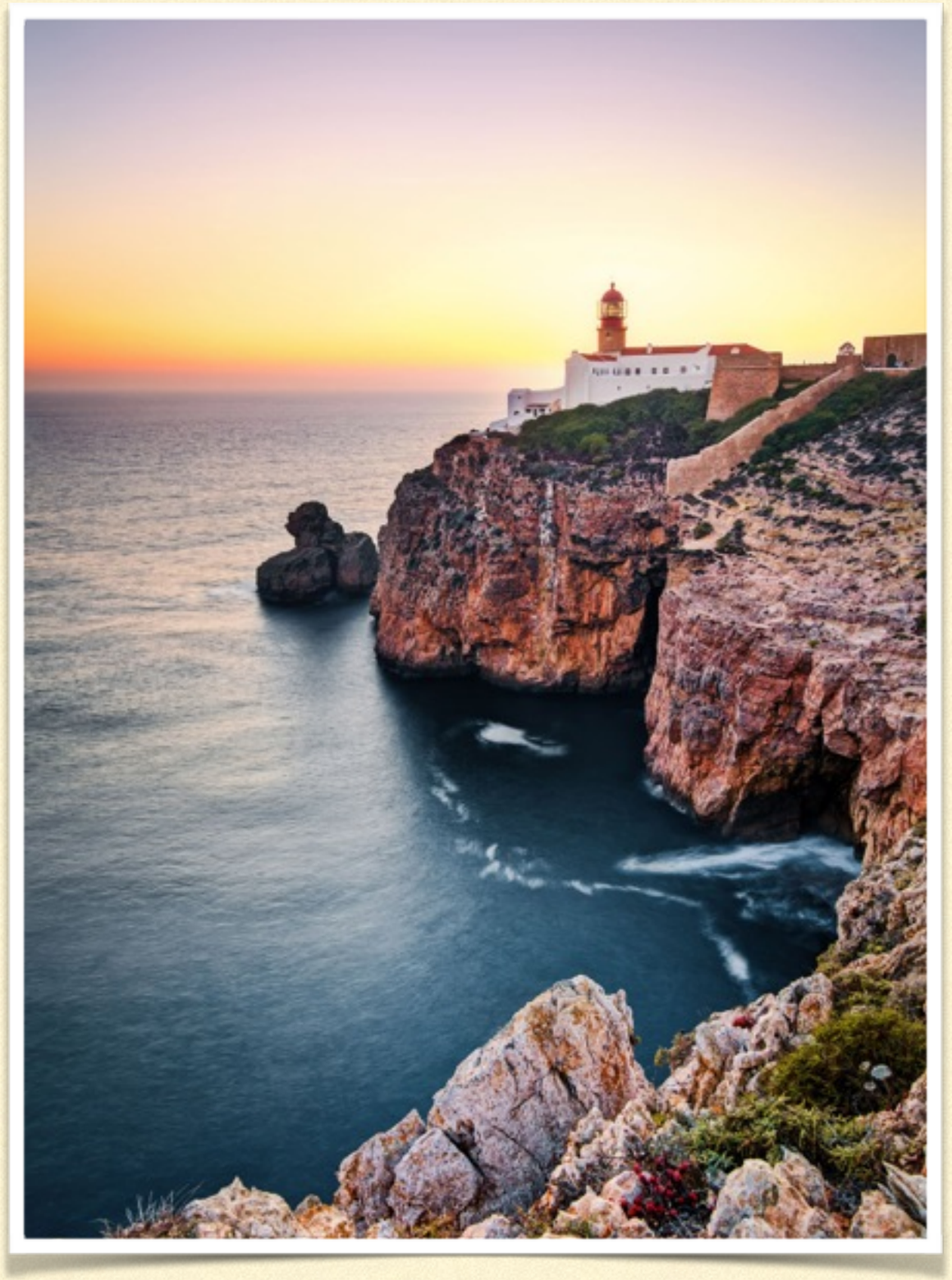
spinlock

mutex

semaphore

monitor

(haoye's join?)



信号量的语义

- 初始化，给与它一个非负数的整数值。
- 运行 P(S)，信号量S的值将被减少1。企图进入临界区段的进程，需要先运行 P(S)。当信号量S减为负值时，进程会被挡住，不能继续；当信号量S不为负值时，进程可以获准进入临界区段。
- 运行V(S)，信号量S的值会被增加1。退出离开临界区段的进程，将会运行V(S)。当信号量S不为负值时，先前被挡住的其他进程，将可获准进入临界区段。

简介 – 历史

[PDF] Co-operating Sequential Processes. F
EW Dijkstra - Programming Languages. Academic Press, New ..., 1968 - alexandria.tue.nl
... CO-OPERATING SEQUENTTAL PROCESSES Prof. Dr. EW Dijkstra Jaren 70 Page 2. Table of Contents. Table of Contents. ... 3.3. The Synchronizing Primitives Applied to the Mutual Exclusion Problem. 4.The General Semaphore. 4.1. Typical Uses of the General Semaphore. 4.2. ...
被引用次数：2326　相关文章　所有 5 个版本　引用　更多

3.1. The Need for a More Realistic Solution.

Let us take the period of time during which one of the processes is in its critical section. We all know, that during that period, no other processes can enter their critical section and that, if they want to do so, they have to wait until the current critical section execution has been completed. For the remainder of that period hardly any activity is required from them: they have to wait anyhow, and as far as we are concerned "they could go to sleep".

简介 – DIJKSTRA的吐槽

- “我们都知道，当一个进程进入它的临界区时，别的进程必须等此进程的临界任务完成，在这一段时间，它们没有任何行动可以执行：它们反正是要等的，还不如让他们’睡一觉’。”
- “当一个积极等待的进程被调入CPU的时候，大量的时间被消耗而这个进程却在原地踏步，就好像这段CPU时间消失了一样。”

LINUX中信号量的实现

- See PV.html

JOS中信号量的实现

- See PV.html