Ubuntu

Our team has decided to pick Ubuntu because it proposes a different approach to its users, unlike its competitors. Ubuntu is not restrictive at all, it is being worked on constantly and kept up to date, is maintained by a community and not by specially designated engineers who are following a business plan enforced by the company, it does not bombard you with ads as the well-known Windows OS or force you to buy an Apple product to enjoy the OS and finally, it is open source.

Ubuntu is a Linux distribution initially released in 2004 and it is based on Debian and composed mostly of free open-source software. Ubuntu is being used for many purposes such as servers, cloud computing and even a personal desktop.

We decided to discuss about the process scheduler, memory management and the I/O scheduler.

Processes and Threads

A process is a series of steps and decisions involved in the way work is completed. A thread of [execution](https://en.wikipedia.org/wiki/Execution_(computing)) is the smallest sequence of programmed instructions .

**Process ID (PID)**

It is the unique identifier of the process. It is used to specify the process to the OS. It does not change until the process termination.

**Credentials**

Also known as User ID (UID) and Group ID (GID). It determines the rights and privilegdes of the process to access system resources.

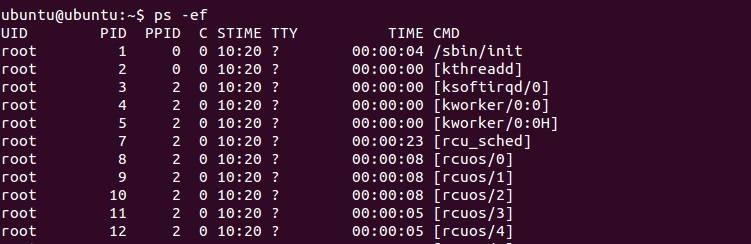
**Commands in CLI** (just a few example)

**PS:** With the appropriate arguments it can show all the running processes, the process runnning from a user

**Kill:** killing a process by PID

**Killall:** killing all processes

**Skill:** process status



The output of PS

Process scheduler

The main problem that arose from the step up to multitasking is dividing the CPU time among different tasks, making way for the scheduler to do its magic. The operating system can have more than one scheduling algorithm because some algorithms may fit some needs and lack to fulfill other needs. Calls to the scheduler take place when it is necessary for a process to wait for an event.

There are two main terms that define how processes communicate and behave to the actions of the scheduler: *cooperative multitasking* and *preemptive multitasking.*

*Cooperative multitasking* happens when one process can give up some CPU power and let another process borrow that CPU power. These scenarios may happen when an I/O device is slow, which means that most of the process’s run time will be spent waiting, leaving only a small portion of run time which will consume CPU time, therefore there is CPU time to give up. However, this has one big flaw: what happens when one process is consuming all the available CPU time, like for instance generating prime numbers using the Sieve of Eratosthenes?

*Preemptive multitasking* comes to our aid and interrupts the process that does not give up the CPU. All UNIX variants, including Linux, use a combination of cooperative and preemptive multitasking.

Due to the preemptive nature of the scheduler, the scheduler implements a scheduling priority and fairness scheme. Whenever this happens, a higher priority process can run before a low priority process. These priorities can be influenced by the users, but ultimately the kernel determines a process’s priority.

Tasks sometimes try to access the same resource. Since UBUNTU Linux are preemptive, if another task with higher priority is ready to use CPU time, the running task will be interrupted even if running in kernel.

The most important aspect in preemptive mechanism is what happens with interrups. In Ubuntu the interrupts are prioritized and seperated into two parts. The top-half one and the bottom-half one. An interrupt form the first class can interrupt one form the second class, but cannot interrupt another of the same class.

Real-time algorithms are also supported, for special time-critical applications that need precise control over the way in which runnable processes are selected for execution.

Three time-sharing algorithms:

* SCHED\_OTHER
* SCHED\_BATCH
* SCHED\_IDLE

**SCHED\_OTHER:** Round-robin method is used, first-comes-first-served (FCFS)

**SCHED\_BATCH:** for "batch" style execution of processes

**SCHED\_IDLE:** floating around

If an unpriviledged SCHED\_IDLE process uses normal kernel functionality, which happens to grab a critical kernel resource such as the root directory's semaphore, and schedules away still holding the semaphore, then there is no guarantee that the task will run again in any

deterministic amount of time - keeping the critical resource potentially forever, there deadlocking every other process that attempts to use that critical resource.

Memory Management

Ubuntu separates physical memory into three different zones:

* zone\_DMA
* zone\_NORMAL
* zone\_HIGHMEM

Zone\_DMA is the zone that includes the first 16MB of physical memory. It is consisted of the DMA-able pages (Direct Access Memory). This happens because certain hardware, Industry Standard Architecture (ISA) devices, can only access the lower 16MB of physical memory.

Zone\_NORMAL is the zone that includes the normally addressable pages up to the first 896 MB of physical memory. The remaining physical memory (>896 MB) is allocated from zone\_HIGHMEM. It is referred as high memory and includes the dynamically mapped pages.

The primary physical-memory manager in the Linux kernel is the page allocator. There is one allocator for each zone, responsible for allocating, freeing all physical pages for each zone, and capable of allocating ranges of physically contiguous pages on request, but even though some allocations may require a page from a particular zone, a normal allocation can come form zone\_DMA or zone\_NORMAL. However, it is preferable to use a page from the particular zone, mostly to save the DMA space, but if needed and memory is low, Kernel can use whichever zone is available and suitable. (). It is important to note that many kernel operations can only take place using zone\_NORMAL so it is the most critical zone about performance.

Ubuntu extends the physical memory of the system using a so called swap partition in the HDD. It is a raw type file-system where pages are swapped when faulting.

Conclusion

Ubuntu provides graphical and terminal interface tools to manage processes. This distribution does not need a specific hardware to run on. As far we know, every PC with almost every kind of hardware config would be able ro run it.

Biography:

http://manpages.ubuntu.com/manpages/hirsute/en/man2/sched\_setscheduler.2.html

https://lwn.net/Articles/3866/

Robert Love, Linux Kernel Development Second Edition

The Linux Programmer’s Toolbox

[Memory management](https://just-more-thoughts.blogspot.com/2011/06/ubuntu-1104-memory-management-36.html)

Daniel Vechiu

Kristof Soos

Fatih Ozkavaf

Laurentiu Bogdan

Sandra- Georgiana Ene