First look on the ps command

As almost everything in Linux, ps is an abbreviation of Process status.

The simplest execution is just without any argument

ps

This way we cannot see much. This way we can see these processes, which are related to this specific session only. This means, that we see the session itself (bash in our case), and processes executed *at this exact moment* by this session (ps in our case) Anyway, we can go through fields now.

* PID - quite obvious, this is the process id.
* TTY - Terminal associated with the process. [Here](https://www.howtogeek.com/428174/what-is-a-tty-on-linux-and-how-to-use-the-tty-command/) is a very detailed reading about TTY.
* TIME - total time of CPU usage
* CMD - the command which is running.

Well, not much, right?

I will say more... These are not all processes associated with this terminal (this, where the ps was executed).

Before wy try more, there is one explanation needed. With ps we can use arguments, quite obvious. But there is a trick. Normally, we use arguments with dash, like ls -l. But ps is different. It accepts argument with and without dash. Try these (and maybe even... try to find the answer what is there:) ):

ps a

ps -a

They are different, correct?

Try ps T

Now we have all processes which has any connection with the current terminal. And also, we see different formatting. Let's ignore that CMD become COMMAND, time changed its granularity. We have new column, STAT. It means state of the process and it is very importnt to understand.

Statuses

This list is copied from manual, which I believe are most important to understand.

* D - uninterruptible sleep (usually IO)
* I - Idle kernel thread
* R - running or runnable (on run queue)
* S - interruptible sleep (waiting for an event to complete)
* T - stopped by job control signal
* t - stopped by debugger during the tracing
* X - dead (should never be seen)
* Z - defunct ("zombie") process, terminated but not reaped by its parent

Some statuses may have the second letter. Let's list the most important

* < - high-priority (not nice to other users)
* N - low-priority (nice to other users)
* s - is a session leader
* l - is multi-threaded
* + - is in the foreground process group

List all processes

To list all processes, use

ps -A

It doesn't give much information, but at least you have all processes listed.

# Most commonly used combinations

Well, this lesson is not going through all arguments and combinations. All is in manual and I am sure you will find your best friend soon. But there are some combinations moslty used when admins run ps command.

ps -ef

Used mostly when someone wants to determine the PID of the process.

ps aux

I think it is the mostly used combination. It shows the most important info, like PID, status and resources usage.

Some admins like to see the hierarchy of processes, therefore they use

ps aux --forest

But in reality, pstree is used by them most often. Let's try

pstree

Personally, I preffer ps aux --forrest from these two. It gives me more details. But to take a quick look on the system and dependencies, pstree might be enough.

## Finding useful information

To be honest... most of people use grep with ps. But ps has enhanced filtering. So why we use grep? Because... it is only one command to lear and we already know it! With great filtering comes great number of argument and switches to learn.

Anyway, let's go through some.

ps -f -u syslog

shows all processes run by user syslog.

ps -f -C cron

shows all processes, where the executable is cron.

ps -f -p 1

shows process with specified PID.

ps -f --ppid 1

show all processes, where parent process has PID 1. About parents, children, etc we will talk in future lesson.

With -p we can specify more PIDs with coma. Something like ps -f -p 2543,8843,3456.

There is much, much more options, views, etc. Here we took a look on the base functionality.

more information is in manual man ps

# Most commonly used combinations

Well, this lesson is not going through all arguments and combinations. All is in manual and I am sure you will find your best friend soon. But there are some combinations moslty used when admins run ps command.

ps -ef

Used mostly when someone wants to determine the PID of the process.

ps aux

I think it is the mostly used combination. It shows the most important info, like PID, status and resources usage.

Some admins like to see the hierarchy of processes, therefore they use

ps aux --forest

But in reality, pstree is used by them most often. Let's try

pstree

Personally, I preffer ps aux --forrest from these two. It gives me more details. But to take a quick look on the system and dependencies, pstree might be enough.

## Finding useful information

To be honest... most of people use grep with ps. But ps has enhanced filtering. So why we use grep? Because... it is only one command to lear and we already know it! With great filtering comes great number of argument and switches to learn.

Anyway, let's go through some.

ps -f -u syslog

shows all processes run by user syslog.

ps -f -C cron

shows all processes, where the executable is cron.

ps -f -p 1

shows process with specified PID.

ps -f --ppid 1

show all processes, where parent process has PID 1. About parents, children, etc we will talk in future lesson.

With -p we can specify more PIDs with coma. Something like ps -f -p 2543,8843,3456.

There is much, much more options, views, etc. Here we took a look on the base functionality.

more information is in manual man ps