

## Linux: Environment variables and command line parameters

### Rapid Q & A:

- Q: Is this a question?
- A: Yes.

### Command line parameters & environment variables:

Command line parameters & environment variables are easy to be retrieved on a Linux system. When your program is started, all the command line parameters and its count are on top of the stack. If you need to retrieve any command line parameter, it should be done as soon as your program has started.

- Parameters count are there as “32 bit” integer on “i386/X86” systems and “64 bit” integer in “X86\_64/AMD64” systems.
- Command line parameters, and environment variables are there as pointer to the parameter string.

To retrieve command line parameters count you just need to:

```
mov     eax,[esp]
mov     [param_count],eax
```

or simply:

```
pop     dword [param_count]
```

Assuming our program was started with 3 command line parameters, named “-help” “about” “abc”, let's get the first 3 parameters:

```
mov     eax,[esp]
mov     [param_count],eax
; get fixed parameter, pointer to executed command string
mov     eax,[esp+$04]
mov     [executed_cmd],eax
; get first actual parameter (-help)
mov     eax,[esp+$08]
mov     [param_1],eax
; get second actual parameter (about)
mov     eax,[esp+$0C]
mov     [param_2],eax
; get third actual parameter (abc)
mov     eax,[esp+$10]
mov     [param_3],eax
; skip null pointer, next is +$18 (end of parameters list)
; get first environment variable
mov     eax,[esp+$18]
mov     [envar_1],eax
; get second environment variable
mov     eax,[esp+$1C]
mov     [envar_2],eax
```

or simply:

```
pop     dword [param_count]
pop     dword [executed_cmd]
pop     dword [param_1]
pop     dword [param_2]
pop     dword [param_3]
; skip null pointer (end of parameters list)
pop     eax
; get first environment variable
pop     dword [envar_1]
; get second environment variable
pop     dword [envar_2]
```

### Notes:

1. On a “X86\_64/AMD64” system you just have to change registers, and “dword” to “qword”.
2. Traditionally, most (if not all) Linux programs include the “executed command string” to the current executable as the first parameter. Because of that, even if your program was started with no parameters, the executed command string will be there as the only 1. Then, if parameter count is 1 it means no actual parameters were given.
3. Both argument and environment variables pointer list on stack ends with a “null” pointer, that determines the end of the parameter list, and then another “null” pointer to signal the end of environment variables list.

Example: <param\_count>,<param\_0>,<param\_1>,<null\_pointer>,<env\_var\_0>,<env\_var\_1>,<null\_pointer>