system / popen / exec

In general: run system commands.

Current recommended way is to use subprocess module. This module intends to **replace several older** modules and functions:

```
os.system
os.spawn*
```

Replacing Older Functions with the subprocess Module

Replacing os.system()

Features:

- possible to execute in shell:
  - all command and params can be specified as string not list
  - multiple commands can be executed at once; shell pipes usage possible
- configurable check for return codes or output

Parameters:

- io-text-encoding

Examples:

```
def run_command(cmd):
    output = Popen(cmd, stdout=PIPE)
    return output.communicate()[0].decode("utf-8").split("\n")
```

see contextlib to catch other components stdout/stderr.

```
import subprocess
meta_data = subprocess.check_output(['netsh', 'wlan', 'show', 'profiles'])
data = meta_data.decode('utf-8', errors ="backslashreplace")
```

```
return subprocess.check_output(
                [self.FW_PRINTENV, '-n', var],
                stderr=subprocess.STDOUT, encoding='ascii').strip()
...
return subprocess.check_call([self.FW_SETENV, var, value])
```