

```

#!/usr/bin/python3
"""
Update system.

This script performs a repair and update.
"""
#####
# Update system.
#
# Written in 2021 by Karel Zimmer <info@karelzimmer.nl>, Creative Commons
# Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0>.
#####

import gettext
import kz_common
import subprocess
import sys

gettext.bindtextdomain('kz', '/usr/share/locale')
gettext.textdomain('kz')
_ = gettext.gettext

#####
# Constants
#####

program_name = 'kz-update'
program_desc = (_('Update system'))
display_name = program_name.replace('kz-', 'kz ')

#####
# Variables
#####

command = ''

#####
# Functions
#####

def perform_repair():
    print(f'* {kz_common.bold}Repair{kz_common.normal} ({display_name})')
    command = 'dpkg --configure --pending'
    perform_command(command)

    command = 'apt-get update --fix-missing'
    perform_command(command)

    command = 'apt-get install --fix-broken'
    perform_command(command)

def perform_update():
    print(f'* {kz_common.bold}Update{kz_common.normal} ({display_name})')
    command = 'apt-get update'
    perform_command(command)

    command = 'apt-get dist-upgrade --yes'
    perform_command(command)

    snap_refresh()

```

```

def perform_command(command):
    print(f'* {command}...')
    kz_common.check_dpkgd_snapd()
    try:
        subprocess.run(f'sudo {command}', shell=True, check=True)
    except Exception as ex:
        print(ex)
        sys.exit(kz_common.error)
    else:
        print()

def snap_refresh():
    print('* snap refresh...')
    kz_common.check_dpkgd_snapd()
    try:
        subprocess.run('sudo snap refresh', shell=True, check=True)
    except Exception as ex:
        print(ex)
    else:
        print()

def perform_cleanup():
    print(f'* {kz_common.bold}Cleanup{kz_common.normal} ({display_name})')
    command = 'apt-get autoclean --yes'
    perform_command(command)

    command = 'apt-get autoremove --yes'
    perform_command(command)

    snap_cleanup()

def snap_cleanup():
    # Start with LANG=en_US.UTF-8 and check for disabled does not always work.
    cmd = '\
snap list --all | \
while read name ver rev trk pub notes; do \
    if [[ $notes == *uitgeschakeld* || $notes == *disabled* ]]; then \
        sudo snap remove "$name" --revision="$rev"; \
    fi \
done'

    print('* snap cleanup...')
    try:
        subprocess.run({cmd}, shell=True, check=True, executable='/bin/bash')
    except Exception as ex:
        print(ex)

def check_reboot():
    try:
        with open('/var/run/reboot-required') as fh:
            print('\n' + _('*** The computer must restart to complete the '
                            'installation of updates ***'))
    except FileNotFoundError:
        pass
    except Exception as ex:
        print(ex)
        sys.exit(kz_common.error)

```

```
#####  
# Main  
#####  
  
kz_common.process_options(program_name, program_desc, display_name)  
kz_common.check_on_ac_power(program_name)  
kz_common.check_user_root(program_name, display_name)  
perform_repair()  
perform_update()  
perform_cleanup()  
check_reboot()
```