
Multitasking with Raspberry Pi



Dogan Ibrahim



elektor

LEARN > DESIGN > SHARE

Preface	10
Chapter 1 • Installing the Raspberry Pi operating system	12
1.1 Overview	12
1.2 Raspbian Buster installation steps on Raspberry Pi 4	12
1.3 Remote access	15
1.4 Using Putty	16
1.4.1 Configuring the Putty	18
1.5 Remote access of the desktop	18
Chapter 2 • Using the Raspberry Pi command line	21
2.1 Overview	21
2.2 Command examples	21
2.3 Installing and removing software.	34
2.4 Shutting down.	34
2.5 Summary	35
Chapter 3 • Process management and resource monitoring on Raspberry Pi	36
3.1 Overview	36
3.2 Foreground and background processing	36
3.3 Task scheduling.	37
3.3.1 Task scheduling management	41
3.3.2 The crontab generator	41
3.4 Running a program or script automatically at system startup	43
3.4.1 Using /etc/rc.local	43
3.4.2 Using crontab	43
3.4.3 Using /etc/init.d.	44
3.5 Resource monitoring	45
3.5.1 Uptime and load average	45
3.5.2 Tasks	45
3.5.3 CPU utilization	46
3.5.4 Memory usage.	46
3.5.5 Process table.	46
3.5.6 Disk usage	49
3.5.7 CPU information	49

3.5.8 Memory use	50
3.5.9 Operating system information	50
3.5.10 USB devices	50
3.5.11 SD card information	51
3.5.12 CPU architecture information	51
3.6 Summary	51
Chapter 4 • Multiprocessing and multithreading	52
4.1 Overview	52
4.2 What is multithreading?	52
4.3 What is multiprocessing?	53
4.4 Differences between multithreading and multiprocessing	53
4.5 Task Scheduling algorithms.	54
4.5.1 Co-operative scheduling	54
4.5.2 Round-robin scheduling	56
4.5.3 Pre-emptive scheduling	57
4.5.4 Scheduling algorithm goals	58
4.5.5 Difference between preemptive and non-preemptive scheduling	59
4.5.6 Some other scheduling algorithms	59
4.5.7 Choosing a scheduling algorithm	60
4.6 Summary	60
Chapter 5 • Raspberry Pi multitasking projects - using the fork()	61
5.1 Overview	61
5.2 Running shell commands from Python	61
5.3 Process forks	62
5.3.1 Project 1 – Two LEDs flashing at different rates	65
5.3.2 Project 2 – Four LEDs flashing at different rates	68
5.3.3 Project 3 – Setting the LED flashing rate from the keyboard	72
5.3.4 Project 4 – Multitasking event counter	74
5.3.5 Project 5 – LED flashing and LED control with a button.	76
5.3.6 Project 6 – Synchronizing the parent and child processes - multitasking event counter	78
5.3.7 Project 7 – Up/down counter.	80

5.4 Summary	83
Chapter 6 • Raspberry Pi multitasking projects - using threads	84
6.1 Overview	84
6.2 Threads	84
6.3 Forking or Threads?	85
6.4 Using threads	85
6.4.1 Project 1 – Two LEDs flashing at different rates	90
6.4.2 Project 2 – Up/down counter	93
6.4.3 Project 3 – Setting the LED flashing rate from the keyboard	97
6.4.4 Project 4 – Setting the LED flashing rate using a button	98
6.4.5 Project 5 – Two-digit 7-segment display seconds counter	101
6.4.6 Project 6 – Two digit 7-segment temperature display	112
6.4.7 Project 7 – Square waveform generator with 7-segment LED display and keyboard	116
6.4.8 Project 8 – Square waveform generator with 7-segment LED display and buttons	120
6.4.9 Project 9 – Four-digit 7-segment display seconds counter	126
6.4.10 Project 10 – Four-digit 7-segment display conveyor belt object counter	131
6.4.11 Project 11 – ON/OFF Temperature controller with LCD	136
6.5 Summary	145
Chapter 7 • Raspberry Pi multitasking projects - using threading	146
7.1 Overview	146
7.2 Threading	146
7.2.1 Thread lock objects	149
7.2.2 Semaphores	150
7.2.3 Events	151
7.2.4 Timer threading	153
7.3 Threading based projects	153
7.3.1 Project 1 – LED flashing counter	154
7.4 Summary	155
Chapter 8 • Using subprocesses	156
8.1 Overview	156

8.2 Subprocesses call	156
8.3 Subprocess run	156
8.4 Subprocess check_call	156
8.5 Subprocess check_output	157
8.6 Subprocess Popen and communicate	157
8.7 Running a Python program	157
8.8 Project 1 – Two LEDs flashing at different rates.	158
8.9 Summary	160
Chapter 9 • Raspberry Pi multitasking projects - using multiprocessing	161
9.1 Overview	161
9.2 Multiprocessing or threading?	161
9.3 How many CPU cores?	162
9.4 Multiprocessing process calls.	162
9.5 Using Events in multiprocessing	163
9.6 Conditions in multiprocessing	163
9.7 Multiprocessing Queues	164
9.8 Sharing data in multiprocessing using Value and Array.	164
9.9 Anonymous Pipes in multiprocessing	164
9.10 Named Pipes	164
9.11 Signals in multiprocessing.	164
9.12 Multiprocessing based projects	164
9.12.1 Project 1 - Two LEDs flashing at different rates	164
9.12.2 Project 2 – Setting the LED flashing rate from the keyboard	166
9.12.3 Project 3 - ON/OFF Temperature controller	167
9.12.4 Project 4 - Metronome	171
9.12.5 Project 5 – Traffic lights controller	175
9.12.6 Project 6 – Ultrasonic car parking aid with buzzer	181
9.12.7 Project 7 – Reaction timer.	188
9.12.8 Project 8 – Stepper motor controller with keyboard	191
9.12.9 Project 9 – Setting the flashing rate of an LED with keypad.	202
9.12.10 Project 10 – Secure door lock with keypad	210
9.12.11 Project 11 – Car park control	214

Appendix A • List of components used in the book	228
Appendix B • Raspberry Pi 4 pin configuration	229
Index	230