تمرین عملی سوم سیستم‌های عامل

برنامه به طور کلی از دو قسمت تشکیل شده است. قسمت اول، سوکت اصلی برنامه است که کلاینت ها در ابتدا به آن پیام می دهند. وقتی این سوکت اصلی، یک کانکشن را قبول کند؛ به کلاینت پیامی مبتنی بر پورت جدیدی که باید ارتباط را از آن ادامه دهد، ارسال کرده(خط ۷۵) و یک ترد جدید می سازد که با این کلاینت ارتباط را ادامه دهد(خط ۷۳ و ۷۸).

```
65  with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
66      s.bind((HOST, PORT))
67      s.listen(10)
68      while True:
69          newSock, addr = s.accept()
70
71          with newSock:
72              token = generate_token()
73              t1 = clientThread(newSock, current_client_port, token)
74              clients.append(t1)
75              a = " {} {}".format(token, current_client_port)
76              newSock.send(a.encode('utf-8'))
77              current_client_port += 1
78              t1.start()
```

قسمت دوم، ادامه ارتباط در هر Thread است. اینجا سرور با port جدید، می تواند سه نوع پیام از کاربر بگیرد – read، که اطلاعات داخل کانال مشترک را می خواند؛ write، که یک داده جدید را در کانال می نویسد و Bye که به ارتباط پایان می دهد. در قسمت write، کلاینت توکن خود و داده موردنظر را ارسال می کند. اگر توکن صحیح بود و هیچ کس دیگر در حال نوشتن روی کانال نبود، دیتای مورد نظر وارد می شود. در غیر این صورت، lock باعث می شود برنامه قبل از cs متوقف شود(خط ۳۱ و ۳۳).

```
26                      if msg == "write":
27                          ncs.send(b"Enter your data!")
28                          c_tok_data = ncs.recv(1024).decode()
29                          if c_tok_data.split()[0] == self.token:
30                              data = ncs.send(b"Wait for data to be written!\n")
31                              lockShared.acquire()
32                              writeShared(c_tok_data.split()[1])
33                              lockShared.release()
34                              ncs.send(b"Data was written!")
35                          else:
36                              ncs.send(b"Token Incorrect!")
```

کلاینت اول، پیام های read و write و bye را با سرور رد و بدل می کند. کلاینت دوم، ورودی اش را متناوبا با فاصله یک ثانیه، تا زمانی که Ctrl+C زده شود، ارسال می کند. (سپس bye ارسال خواهد شد.)