	Работа с сетью, сокеты • Что такое сокеты? • Зачем нужны сокеты? • Программа клиент-сервер
In []:	
	Сокет в ОС Linux - это объект уровня ядра. T.e. python процесс, при создании сокета и вызова функций осуществяет системные вызовы. Ядро ОС возвращает результаты работы системных вызовов python процессу.
In []:	# Cosounac Concina; CCpocp
	<pre>import socket # https://docs.python.org/3/library/socket.html sock=socket.socket(socket.AF_INET, socket.SOCK_STREAM)</pre>
	sock.bind(("127.0.0.1", 10001)) # max port 65535 sock.listen(socket.SOMAXCONN)
	<pre>conn, addr = sock.accept() while True: data = conn.recv(1024) if not data:</pre>
	<pre>break # process data print(data.decode("utf8"))</pre>
	<pre>conn.close() sock.close()</pre>
In []:	# создание сокета, клиент import socket
	<pre>sock = socket.socket() sock.connect(("127.0.0.1", 10001)) sock.sendall("ping".encode("utf8"))</pre>
	sock.close() # более короткая запись
	<pre>sock = socket.create_connection(("127.0.0.1", 10001)) sock.sendall("ping".encode("utf8")) sock.close()</pre>
In []:	# создание сокета, контекстный менеджер # сервер
	<pre>import socket with socket.socket() as sock: sock.bind(("", 10001))</pre>
	sock.listen() while True:
	<pre>conn, addr = sock.accept() with conn: while True: data = conn.recv(1024)</pre>
	<pre>if not data: break print(data.decode("utf8"))</pre>
	# клиент import socket
	<pre>with socket.create_connection(("127.0.0.1", 10001)) as sock: sock.sendall("ping".encode("utf8"))</pre>
	Таймауты и обработка сетевых ошибок
	• connect timeout и read timeout, в чем разница? • обработка ошибок
In []:	# создание сокета, таймауты и обработка ошибок # сервер import socket
	<pre>with socket.socket() as sock: sock.bind(("", 10001)) sock.listen()</pre>
	<pre>while True: conn, addr = sock.accept() conn.settimeout(5) # timeout := None 0 gt 0</pre>
	<pre>with conn: while True: try: data = conn.recv(1024)</pre>
	except socket.timeout: print("close connection by timeout") break
	<pre>if not data: break print(data.decode("utf8"))</pre>
In []:	# создание сокета, таймауты и обработка ошибок # клиент
	<pre>import socket with socket.create_connection(("127.0.0.1", 10001), 5) as sock: # set socket read timeout</pre>
	<pre>sock.settimeout(2) try: sock.sendall("ping".encode("utf8"))</pre>
	<pre>except socket.timeout: print("send data timeout") except socket.error as ex: print("send data error:", ex)</pre>
	Одновременная обработка нескольких соединений
	Одновременная обработка нескольких соединений • Как обработать несколько соединений одновременно? • Что использовать процессы или потоки? • Рассмотрим примеры обработки сетевых запросов
In []:	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно
	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов
	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock: sock.bind(("", 10001)) sock.listen() while True: conn, addr = sock.accept()
	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock:
	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock:
	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock: sock.bind(("", 10001)) sock.listen() while True: conn, addr = sock.accept() print("connected client:", addr) # процесс или поток для обработки соединения with conn: while True: data = conn.recv(1024) if not data: break print(data.decode("utf8")) Какие преимущества дает создание процессов?
In []:	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock: sock.bind(("", 10001)) sock.listen() while True: conn, addr = sock.accept() print("connected client:", addr) # процесс или поток для обработки соединения with conn: while True: data = conn.recv(1024) if not data: break print(data.decode("utf8"))
In []:	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработко нескольких соединений одновременно import socket with socket.socket() as sock:
In []:	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock:
In []:	 Как обработать несколько соединений одновременно? Что использовать процессы или потоки? Рассмотрим примеры обработки сетевых запросов # обработка нескольких соединений одновременно import socket with socket.socket() as sock: sock.bind((", 1808))) sock.listen() while True:
In []:	 • Κακ οδραδοτατь несколько соединений одновременно? • Что использовать προцессы или потоки? • Рассмотрим примеры обработки сетевых запросов # οδραδοπικ νεскальких соединений облабромского import socket with socket.socket() as sock. socket() as sock. socket.inter() sock.bind(("", 19001)) sock.inter() while True:
In []:	• Κακ οδραδοτατь несколько соединений одновременно? • Что использовать процессы или потоия? • Рассмотрим примеры обработки сетевых запросов # οδραδοπαι нескольких соединений одновремения import socket with socket.socket() as sock: sock.bind((", 18001)) sock.listen() while True: conn, addr = sock.accept() print("connected client:", addr) # προωσει υπι ποπου Δαν οδραδοπου coeduнения with conn: while True: data = conn.recv(1824) if not data: break print(data.decode("urf8")) Kaknen premyugerous деет создание процессои? Можно утилимировать исе жары СРО. Вазор боск = это слинком такелаю операцие. Использование потоков ограничено БТІ и одним процессом. Ми можем ксчернать 100% (ке) и одном вдре, попражно будет работать неэфективно. Как поведет себя ОС при Большой раскод памяти. # обработка нескольких соединемий одмобремению, лотоков?
In []:	 Как обработать несколько соединений одновременно? Что использовать процессы или потожи? Рассмотриии примеры обработки сетевых запросов ф обрабомо мескольких соединений обмобременно Limport socket with socket, socket() as sock: sock bind((***, 1880))) until Liston() print(*connected < lient(***, addr) print(***, addr)
In []:	- Και οδραθοντατι μοεκοπικό εκομικό εκριμανική ομπαραμανική ο - Υπου αντοπρουστική προμεστική εντοποική? - Ρασεκοτρική προμεστική εκριμανική
In []:	• Kax oбработать нескольмо соединений одновременно? • Что использовать процессы или потоки? • Рассмотрим примеры обработки сегевых запросов # обработата исскольных соединений одновременью Import socket sock bunu((***, 19991)) sock issen(**, 19991)) sock issen(**, 19991)) sock issen(**, 19991) print(**connected client;**, adde) print(**connected client;**, adde) print(**connected client;**, adde) # nepeace или полок дол обработими соединения while true; data - conn.recv(1924) if net data:: break print(\$tata decode(*utf8**)) Kawne правичущества двет создаваем процессов? Кизно упиливаровать не пилы КВ. Килоп бой - за сенишко такогом оперция. Килоп бой - за сенишко такогом оперция. Килоп бой - за сенишко такогом оперция. Киспользование потокое ограничено СТ и однам процессом. Ры можем иссервать 1905 (PU из одном вдре, приграмки будет рыбитеть инхуфистивии. ККК гоездет себа ОС пре большом кол-ве процессов. # обрабовия мескольных соединемий однобравенно, колоки inport toreating def process request(cons, adde); print(**connected client;**, 334-) with socket socket() as sock: sock binu(***, 19891)) with socket-socket() as sock: sock binu(***, 19891) with socket-socket() as sock: sock binu(****, 19891) with socket-socket() as sock: sock binu(*****, 19891) with socket-socket() as sock: sock binu(************************************
In []:	• Kak objectorats Herotomoto coequeenum quipospemenum? • His outcomisowan in poqueccu или по том? • Paccoccopy my proposo objector occessor aarpocos # objectomotomotomotomotomotomotomotomotomotom
In []: In []:	* Kas oбработать насколько соединений одновременно? • Tru Uncordsoports процессы или потоля? * Edipodomou rescolessos coedunenud obeddessenes # ofpodomou rescolessos coedunenud obeddessenes # ofpodomou rescolessos coedunenud obeddessenes # ofpodomou rescolessos coedunenud obeddessenes # of podomou rescolessos coedunenud obeddessenes # of podomou rescolessos coedunenud obeddessenes # of podomou rescolessos coedunenud # of podomou rescolessos # of podomou res
In []: In []:	Run objections in inconstruct configuration forced in the forced in Paramateria protection of the forced in the forced in Paramateria protection of the forced in the for
In []: In []:	Exc of particulars are exceeded as a particular and account of particular account of particular and account of particular
In []: In []:	Kas ofconforms recommon conformed and processor and proce
In []: In []:	None objection Author of the content of the con
In []: In []:	* Note of particulars - concentral conjunction of conjunction of conjunction of conjunction in plantaces with interest. * Processor of management is entitlement closely assessed interest to extend the confunction of the conjunction of the c
In []: In []:	- Non-Oppolicians Necconico Congresenti Capcoperacieno 7 - Uno Commissata notacco del normali del production del normali del productiono del normali del productiono del normali del productiono del normali del productiono del normali d
In []: In []:	Note the Content of
In []: In []:	- Post Confidence in Section of Confidence Confidence in
In []: In []:	• To displace in the control of the
In []: In []:	- No confidence in consistence of control o
In []: In []:	F. Note Control C
In []: In []:	Fig. 6 (Control of the Control of Control o
In []: In []:	Book Confidence with the Confidence of the
In []: In []:	Policy Control Control (Control Control C