Экзамен ОВС

Лисенкова Е. А., ПИ19-2

Билет 1.

**Теория 1. Особенности процессорных архитектур. CISC и RISC архитектура. Их краткая характеристика.**

Исторически сложилось, что существуют много процессоров и много архитектур. Но многие архитектуры имеют схожести. Специально для этого появились "Группы" архитектур типа RISC, CISC, MISC, OISC (URISC). Кроме того они могут иметь разные архитектуры адресации памяти (фон Неймана, Гарвард). У каждого процессора есть своя архитектура. Например большинство современных архитектур это RISC (ARM, MIPS, OpenRISC, RISC-V, AVR, PIC\*\* и т.д.), но есть архитектуры которые выиграли просто за счет других факторов (Например удобство/цена/популярность/etc) Среди которых x86, x86-64 (Стоит отметить, что x86-64 и x86 в последних процессорах используют микрокод и внутри них стоит RISC ядро), M68K.  Поподробнее рассмотрим CISC и RISC архитектуры.

**CISC** (англ. Complex Instruction Set Computer — «компьютер с полным набором

команд») — тип процессорной архитектуры, в первую очередь, с нефиксированной

длиной команд, а также с кодированием арифметических действий в одной команде и

небольшим числом регистров, многие из которых выполняют строго определенную

функцию.

Самый яркий пример CISC архитектуры — это x86 (он же IA-32) и x86\_64 (он же

AMD64).

В CISC процессорах одна команда может быть заменена ей аналогичной, либо группой

команд, выполняющих ту же функцию. Отсюда вытекают плюсы и минусы

архитектуры: высокая производительность благодаря тому, что несколько команд

могут быть заменены одной аналогичной, но большая цена по сравнению с RISC

процессорами из-за более сложной архитектуры, в которой многие команды сложнее

раскодировать.

CISC — Complex Instruction Set Computer — ее особенность в увеличенных количествах действий за инструкцию. Таким образом можно было теоретически увеличить производительность программ за счет увеличения сложности компилятора. Но по факту у CISC плохо были реализованы некоторые инструкции т.к. они редко использовались, и повышение производительности не было достигнуто. Особенностью этой группы является еще огромная разница между архитектурами. И несмотря на названия были архитектуры с маленьким количеством инструкций.

**RISC** (англ. Reduced Instruction Set Computer — «компьютер с сокращённым набором

команд») — архитектура процессора, в котором быстродействие увеличивается за счёт

упрощения инструкций: их декодирование становится более простым, а время

выполнения — меньшим. Первые RISC-процессоры не имели даже инструкций

умножения и деления и не поддерживали работу с числами с плавающей запятой.

По сравнению с CISC эта архитектура имеет константную длину команды, а также

меньшее количество схожих инструкций, позволяя уменьшить итоговую цену

процессора и энергопотребление, что критично для мобильного сегмента. У RISC

также большее количество регистров.

Примеры RISC-архитектур: PowerPC, серия архитектур ARM (ARM7, ARM9, ARM11,

Cortex).

В общем случае RISC быстрее CISC. Даже если системе RISC приходится выполнять 4

или 5 команд вместо одной, которую выполняет CISC, RISC все равно выигрывает в

скорости, так как RISC-команды выполняются в 10 раз быстрее.

**Теория 2.** **Виды обеспечения вычислительных систем. Определения. Примеры**.

Существуют следующие основные виды обеспечения вычислительных систем:

* программное,
* техническое(аппаратное),
* математическое,
* информационное,
* лингвистичекое,
* операционное,
* методическое.

Основными из них являются **программное** и **техническое**. Их и рассмотрим поподробнее.

1. К *аппаратному* обеспечению вычислительных систем относятся устройства и приборы, образующие аппаратную конфигурацию. Современные компьютеры и вычислительные комплексы имеют блочно-модульную конструкцию — аппаратную конфигурацию, необходимую для исполнения конкретных видов работ, можно собирать из готовых узлов и блоков. *Техническое(аппаратное)* обеспечение включает компьютеры и логические устройства, внешние устройства и диагностическую аппаратуру, энергетическое оборудование, батареи и аккумуляторы.

2.Программы — это упорядоченные последовательности команд. Конечная цель любой компьютерной программы — управление аппаратными средствами. Даже если на первый взгляд программа никак не взаимодействует с оборудованием, не требует никакого ввода данных с устройств ввода и не осуществляет вывод данных на устройства вывода, все равно ее работа основана на управлении аппаратными устройствами компьютера.

*Программное* обеспечение – программа или множество программ, используемых для управления компьютером. Вычислительные машины в настоящее время не могут

работать без соответствующего программного обеспечения. То есть для того, чтобы

компьютер выполнял те или иные задачи, на него должны быть установлены

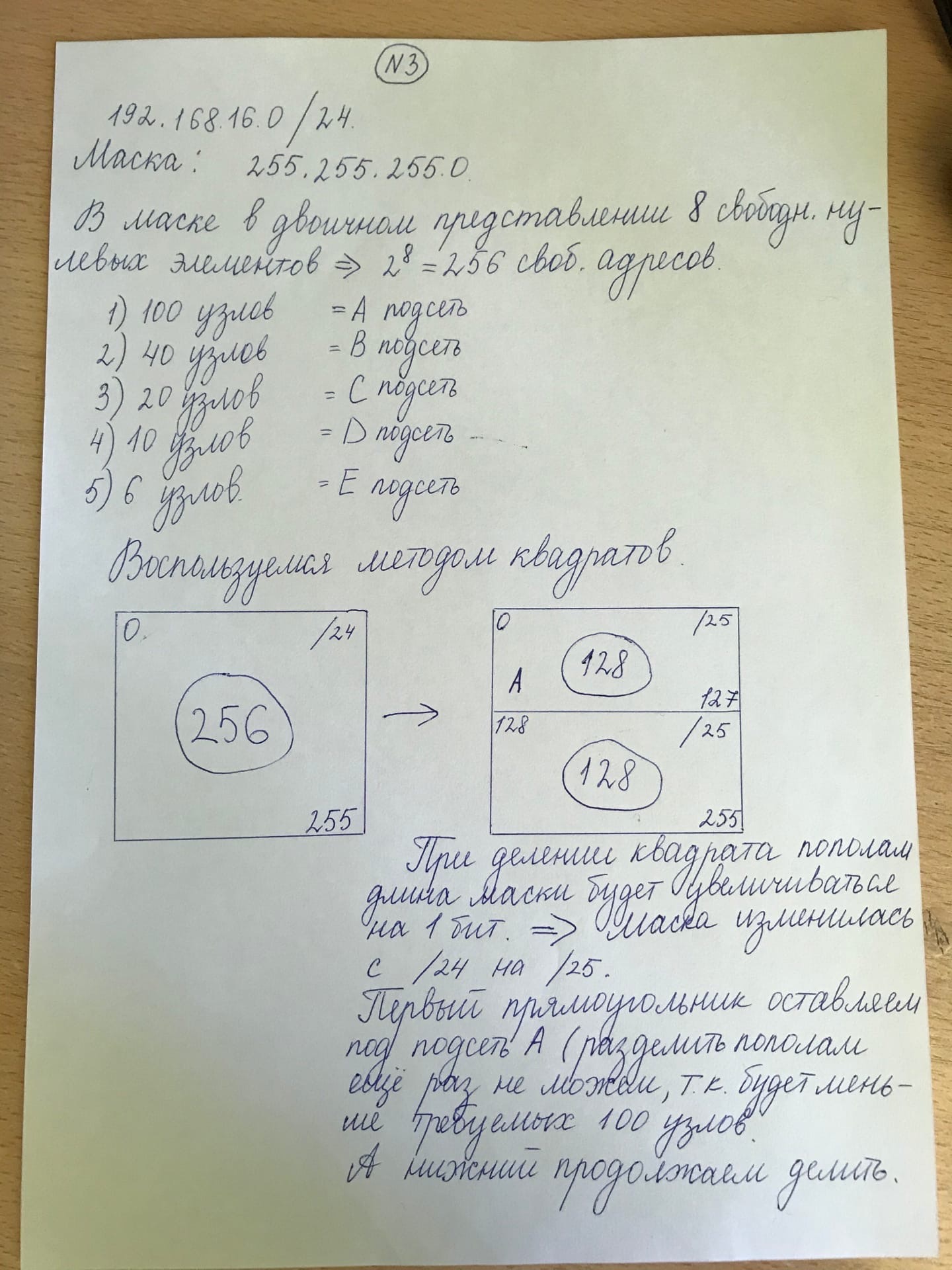
программы, содержащих алгоритмы выполнения этих задач.

Программное и аппаратное обеспечение в компьютере работают в неразрывной связи и в непрерывном взаимодействии. Несмотря на то что мы рассматриваем эти две категории отдельно, нельзя забывать, что между ними существует диалектическая связь, и раздельное их рассмотрение является по меньшей мере условным.

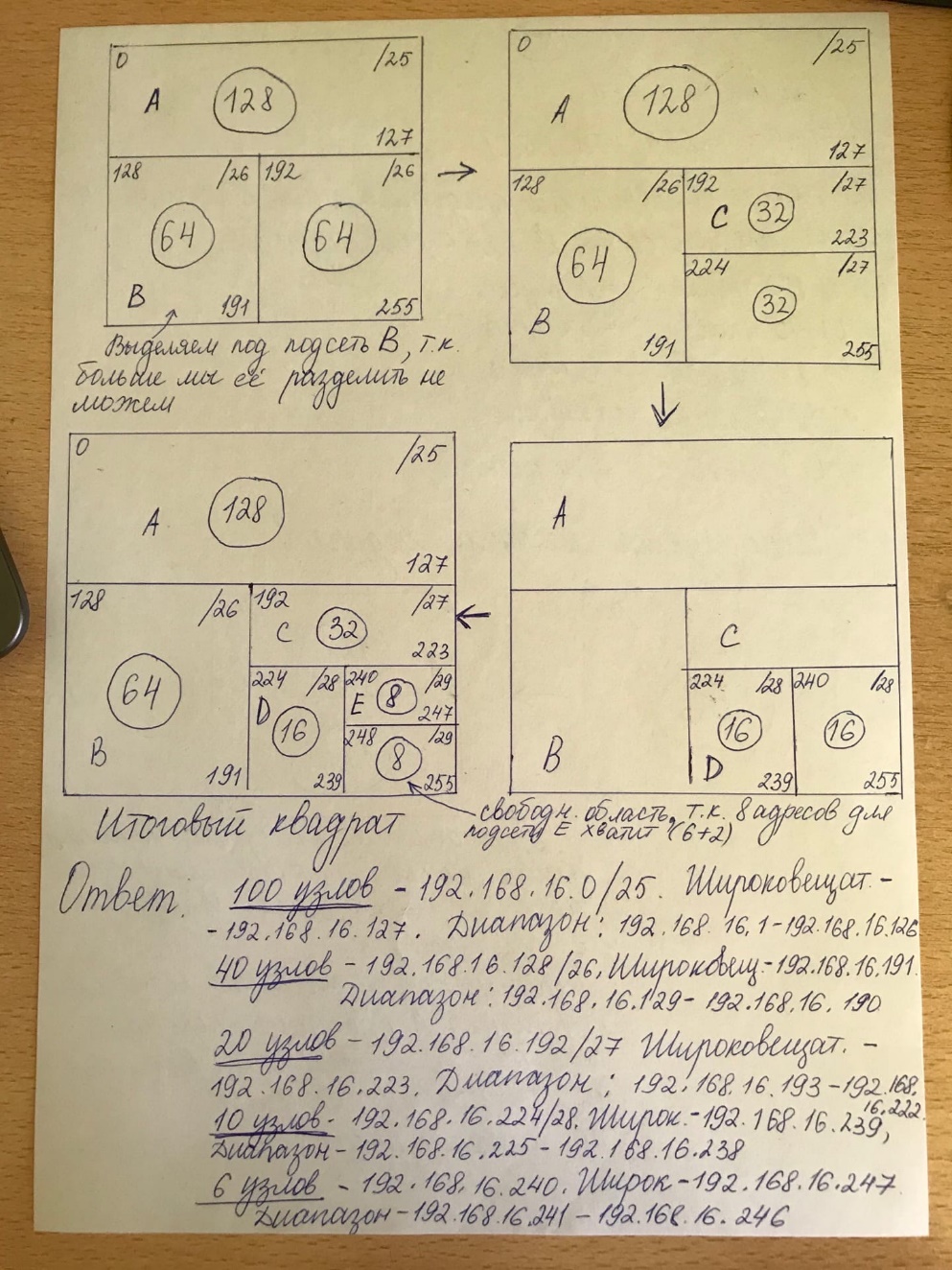
**Задача 3.**

Разделить IP-сеть на подсети в соответствии с номером задания. Для каждой подсети указать широковещательный адрес. 192.168.16.0/24, 5 подсетей с 100, 20, 10, 6 и 40 узлами . (30 баллов).

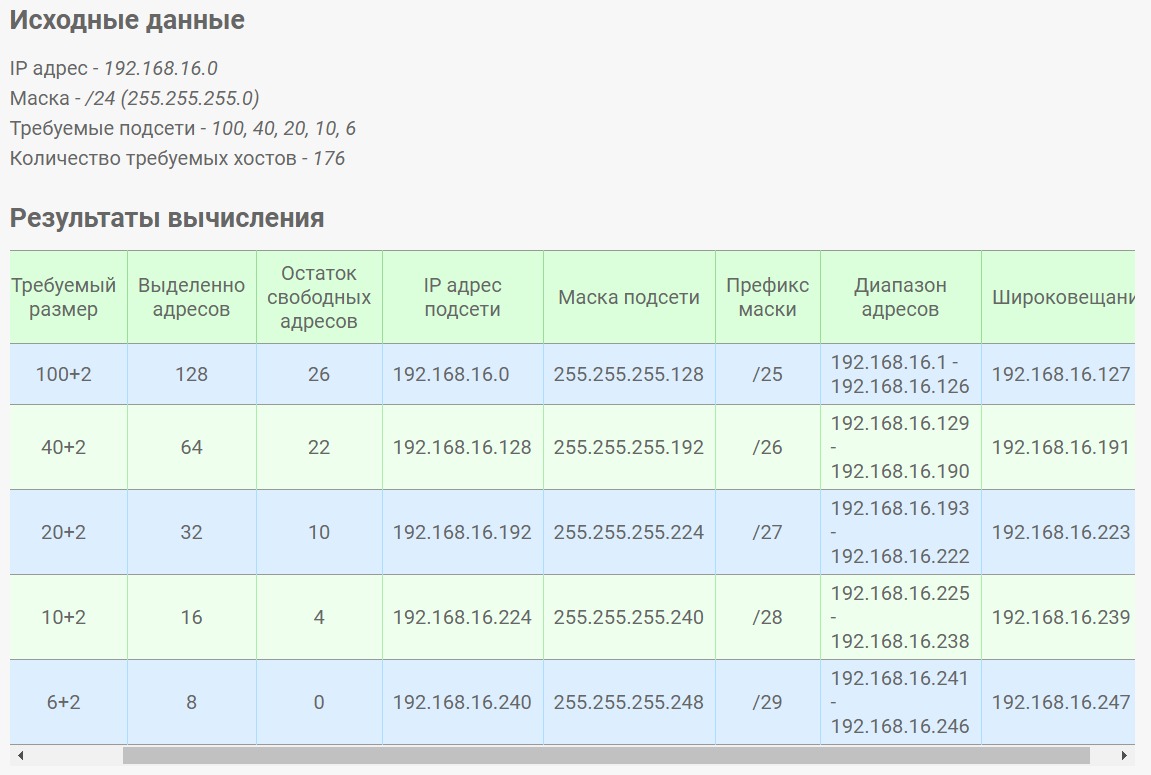
См. ниже



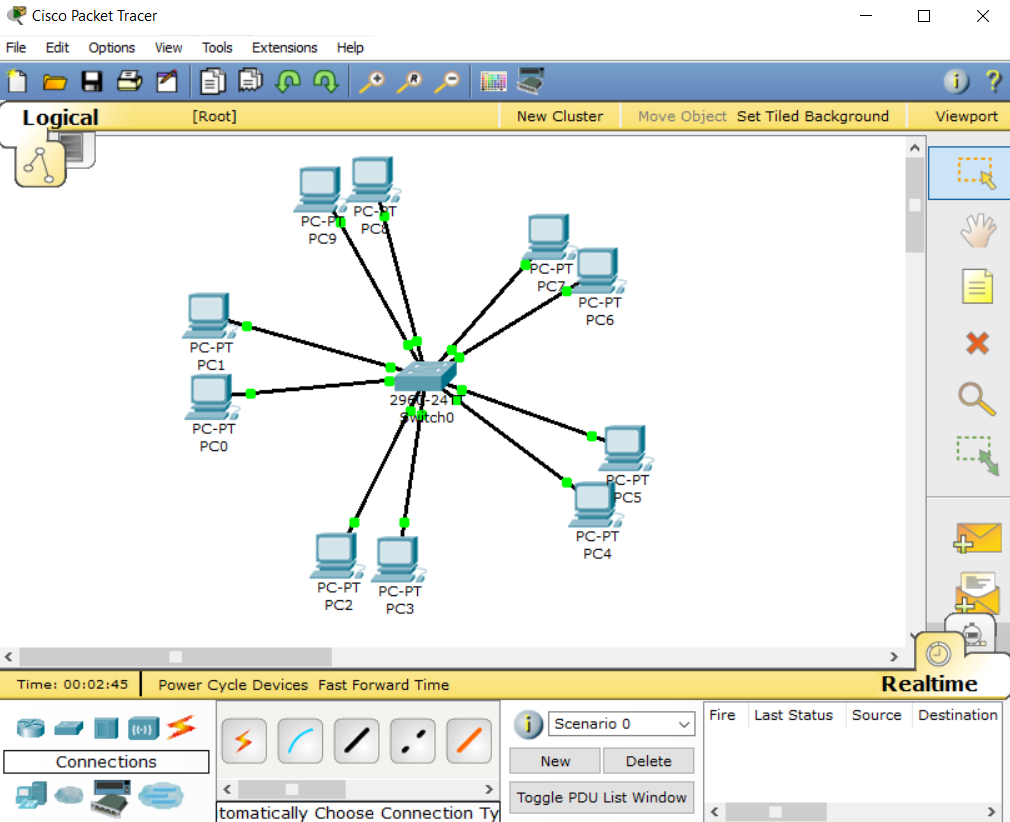
2 страница на след. стр.



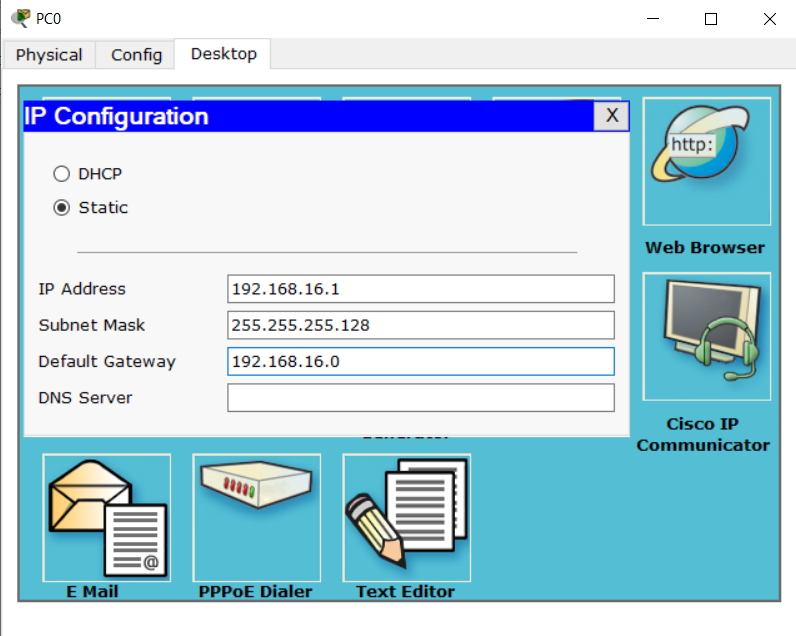
Скрин из онлайн калькулятора:

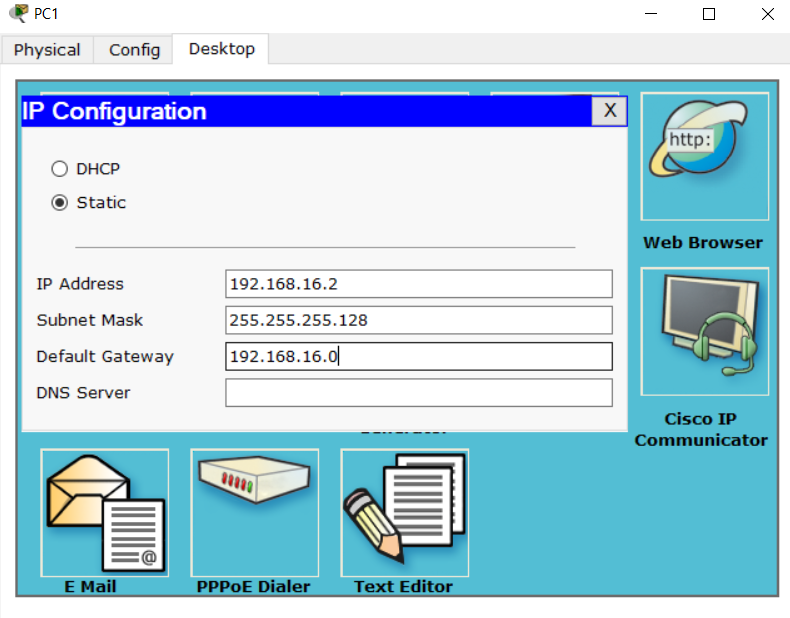


Работа в Cisco:

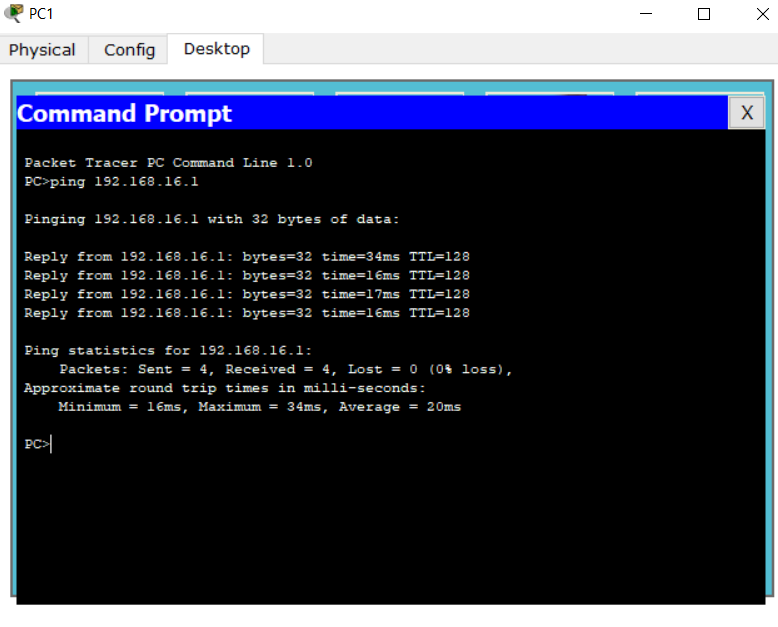


Компьютеры одной сети:



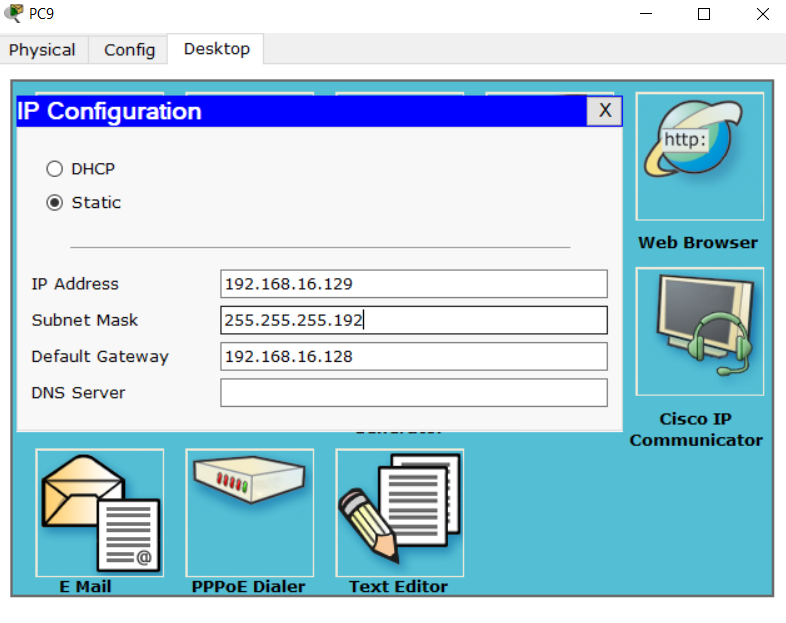


Пингуем их:



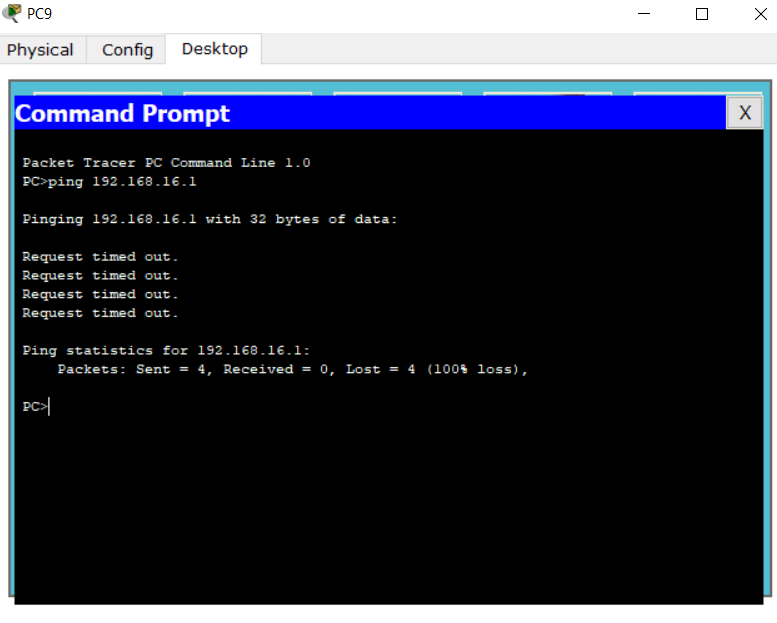
Вывод: компьютеры одной сети пингуются.

Теперь попробуем пропинговать с компьютером из другой сети.



Вот компьютер из другой сети.

Пингуем его.



Вывод: Компьютеры разных сетей не пингуются!

Итог: компьютеры одной сети видят друг друга, Разных- нет.