



Universidad Autónoma de Chiapas

Facultad: Contaduría y Administración

Licenciatura: Ingeniería en Desarrollo y Tecnologías de Software

Conmutadores y Redes inalámbricas

“Act. 2.4 Configuración de DHCP en Router Cisco usando
GNS3.”



Alumno: Nango Ponce Manuel de Jesus

Grupo: 7M

Matrícula: A200338

Docente: Luis Gutiérrez Alfaro

Tuxtla Gutiérrez, Chiapas
09/09/2023

ÍNDICE:

INTRODUCCIÓN.....	2
1. Arquitectura de Computadoras.....	2
2. Lenguaje Ensamblador.....	4
¿Cómo se relaciona el lenguaje ensamblador con la arquitectura del procesador?.....	4
CONCLUSIÓN.....	5
FUENTES DE INFORMACIÓN.....	5

INTRODUCCIÓN

La generación de código es una de las etapas fundamentales en la creación de un lenguaje de programación. En esta etapa, el código fuente escrito en un lenguaje de alto nivel se convierte en código de máquina ejecutable por la computadora. La arquitectura de la computadora juega un papel crucial en la generación de código, ya que es la base sobre la cual se construye el software. En este informe, se explorará en profundidad la arquitectura de la computadora y su relación con la generación de código.

1. Arquitectura de Computadoras

La arquitectura de la computadora se refiere a la estructura interna de una computadora y cómo se comunica con los dispositivos externos. La arquitectura de la computadora se divide en dos partes principales: hardware y software. El hardware se refiere a los componentes físicos de una computadora, como la CPU, la memoria y los dispositivos de entrada/salida. El software se refiere a los programas que se ejecutan en una computadora.

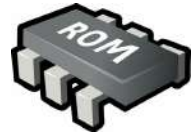
En cuanto a la arquitectura del procesador, hay dos tipos principales: RISC (Reduced Instruction Set Computing) y CISC (Complex Instruction Set Computing). Los procesadores RISC tienen un conjunto reducido de instrucciones que son más simples y fáciles de ejecutar. Los procesadores CISC tienen un conjunto más amplio de instrucciones que son más complejas pero pueden realizar más tareas en una sola instrucción.

A continuación, se presentan algunos de los principales componentes de la arquitectura de la computadora:

- **Procesador:** El procesador, también conocido como Unidad Central de Procesamiento (CPU), es el componente central de la arquitectura de la computadora. Es el encargado de realizar las operaciones de cálculo y lógica necesarias para ejecutar programas y procesar datos. Los procesadores modernos tienen múltiples núcleos que les permiten realizar varias tareas simultáneamente.



- **Memoria:** La memoria es el componente de la arquitectura de la computadora que almacena los datos y programas que la computadora está utilizando en un momento dado. La memoria se divide en dos tipos principales: la memoria RAM (Random Access Memory) y la memoria ROM (Read-Only Memory). La memoria RAM es volátil, lo que significa que pierde los datos cuando se apaga la computadora, mientras que la memoria ROM es no volátil y conserva su contenido incluso cuando la computadora está apagada.



- **Bus de datos:** El bus de datos es el medio por el cual se transmiten los datos entre los componentes de la computadora. El bus de datos está compuesto por una serie de líneas de comunicación que permiten la transferencia de datos en ambas direcciones.



- **Dispositivos de entrada/salida:** Los dispositivos de entrada/salida (E/S) son los componentes de la arquitectura de la computadora que permiten la comunicación entre la computadora y el usuario o entre la computadora y otros dispositivos externos. Los dispositivos de entrada incluyen el teclado, el ratón y otros dispositivos de entrada táctil, mientras que los dispositivos de salida incluyen la pantalla, las impresoras y los altavoces.



- **Periféricos:** Los periféricos son componentes externos de la computadora que se utilizan para ampliar las capacidades de la computadora. Los periféricos pueden ser dispositivos de entrada/salida, como discos duros externos o unidades flash USB, o dispositivos de comunicación, como routers o módems.

En resumen, la arquitectura de la computadora es un conjunto de componentes físicos y lógicos que trabajan juntos para realizar tareas y procesar datos. **La generación de código depende de la arquitectura de la computadora, ya que el software debe ser diseñado para funcionar de manera eficiente con los componentes de hardware disponibles.**

2. Lenguaje ensamblador

El lenguaje ensamblador es un lenguaje de programación de bajo nivel que se utiliza para escribir programas informáticos que se ejecutan directamente en la arquitectura de la computadora. El lenguaje ensamblador es una representación más directa del código máquina específico para cada arquitectura de computadoras legible por un programador.

La generación de código objeto es la última etapa del proceso de compilación y se encarga de generar código objeto que pueda ser ejecutado por una computadora. El ensamblador es un traductor de un código de bajo nivel a un código ejecutable directamente por la máquina para la que se ha generado.

El lenguaje ensamblador utiliza código mnemotécnico para formular instrucciones básicas que son interpretadas por los computadores, procesadores, controladores y cualquier circuito integrado que sea programable. Muchos ensambladores tienen macros incorporadas (o predefinidas) para las llamadas de sistema y otras secuencias especiales de código, tales como la generación y el almacenamiento de los datos realizados a través de avanzadas operaciones bitwise y operaciones booleanas usadas en juegos, software de seguridad, gestión de datos y criptografía.

¿Cómo se relaciona el lenguaje ensamblador con la arquitectura del procesador?

El lenguaje ensamblador se relaciona con la arquitectura del procesador porque el lenguaje ensamblador es específico para cada arquitectura de procesador. El lenguaje ensamblador es una representación más directa del código máquina específico para cada arquitectura de computadoras legible por un programador.

El lenguaje ensamblador utiliza instrucciones que son específicas para la arquitectura del procesador. Cada procesador tiene su propio conjunto de instrucciones que se pueden utilizar en el lenguaje ensamblador. Por lo tanto, el lenguaje ensamblador es específico para cada arquitectura de procesador.

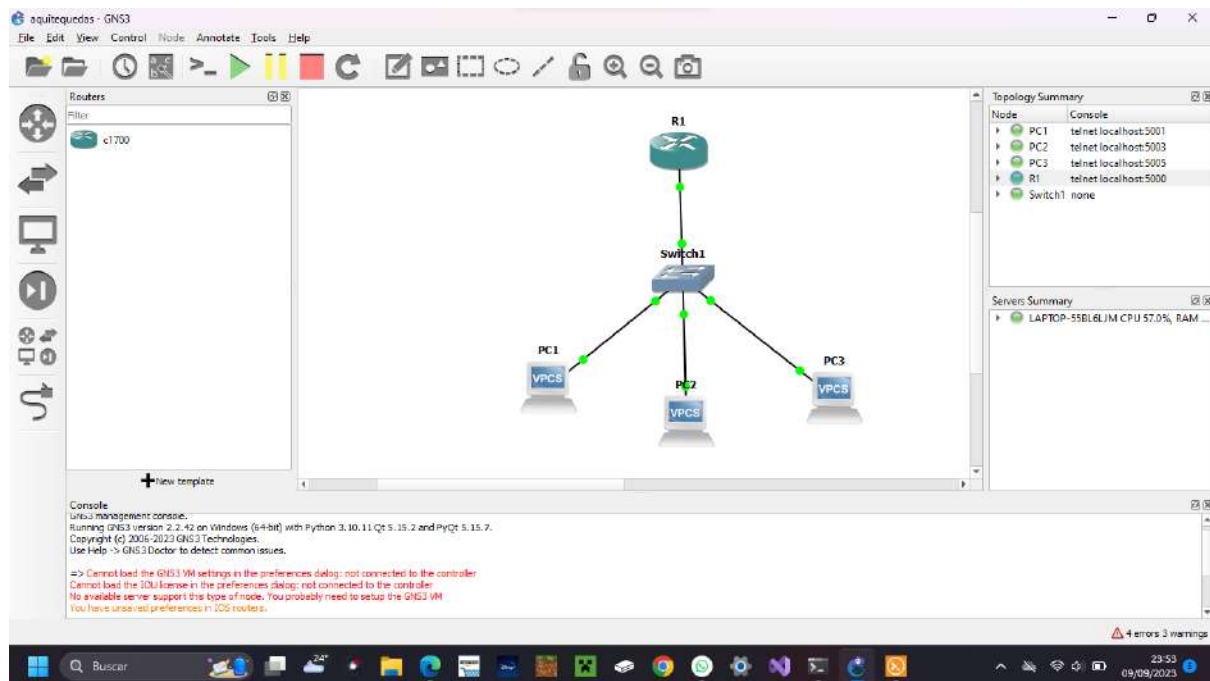
CONCLUSIÓN:

En conclusión, la arquitectura de computadora es el diseño conceptual y la estructura fundamental de un sistema informático. La arquitectura de computadora se divide en tres componentes principales: hardware, sistema operativo y software de aplicación. El lenguaje ensamblador es un lenguaje de programación de bajo nivel que se utiliza para escribir programas informáticos que se ejecutan directamente en la arquitectura de la computadora. El lenguaje ensamblador es una representación más directa del código máquina específico para cada arquitectura de computadoras legible por un programador.

FUENTES DE INFORMACIÓN:

1. Unidad IV: Generación de código objeto 4.1 Registro - ITPN. (s.f.). Recuperado 8 de mayo de 2023, de <http://www.itpn.mx/recursosisc/7semestre/leguajesyautomatas2/Unidad%20IV.pdf>
2. Sintaxis Lenguaje Ensamblador – Arquitectura de Computadoras. (2016, 11 mayo). Recuperado 8 de mayo de 2023, de <https://jrgsac4a.wordpress.com/2016/05/11/lenguaje-ensamblador/>
3. Lenguaje ensamblador - Wikipedia, la enciclopedia libre. (s.f.). Recuperado 8 de mayo de 2023, de https://es.wikipedia.org/wiki/Lenguaje_ensamblador
4. Lenguaje Ensamblador: ¿Qué es y cómo funciona? (s.f.). Recuperado 8 de mayo de 2023, de <https://open-bootcamp.com/aprender-programar/lenguaje-ensamblador>

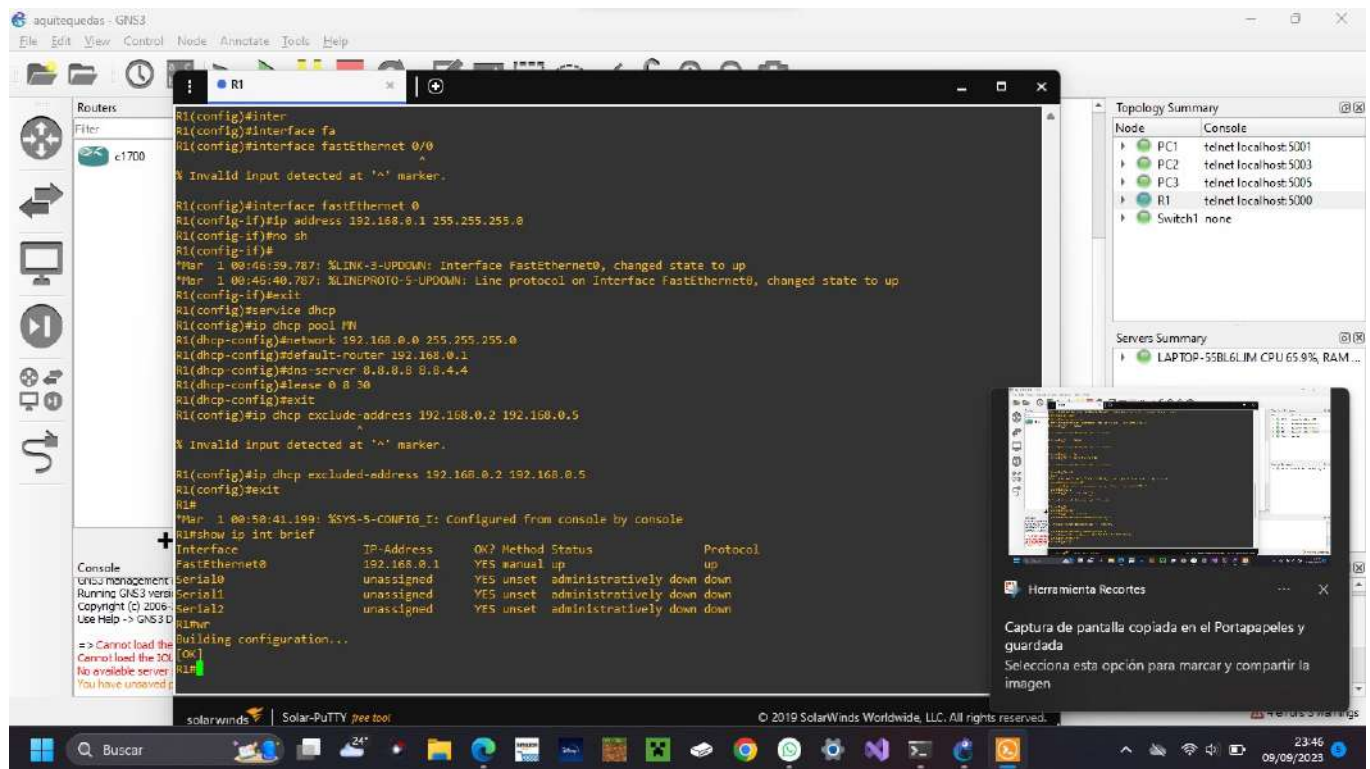
Topología:



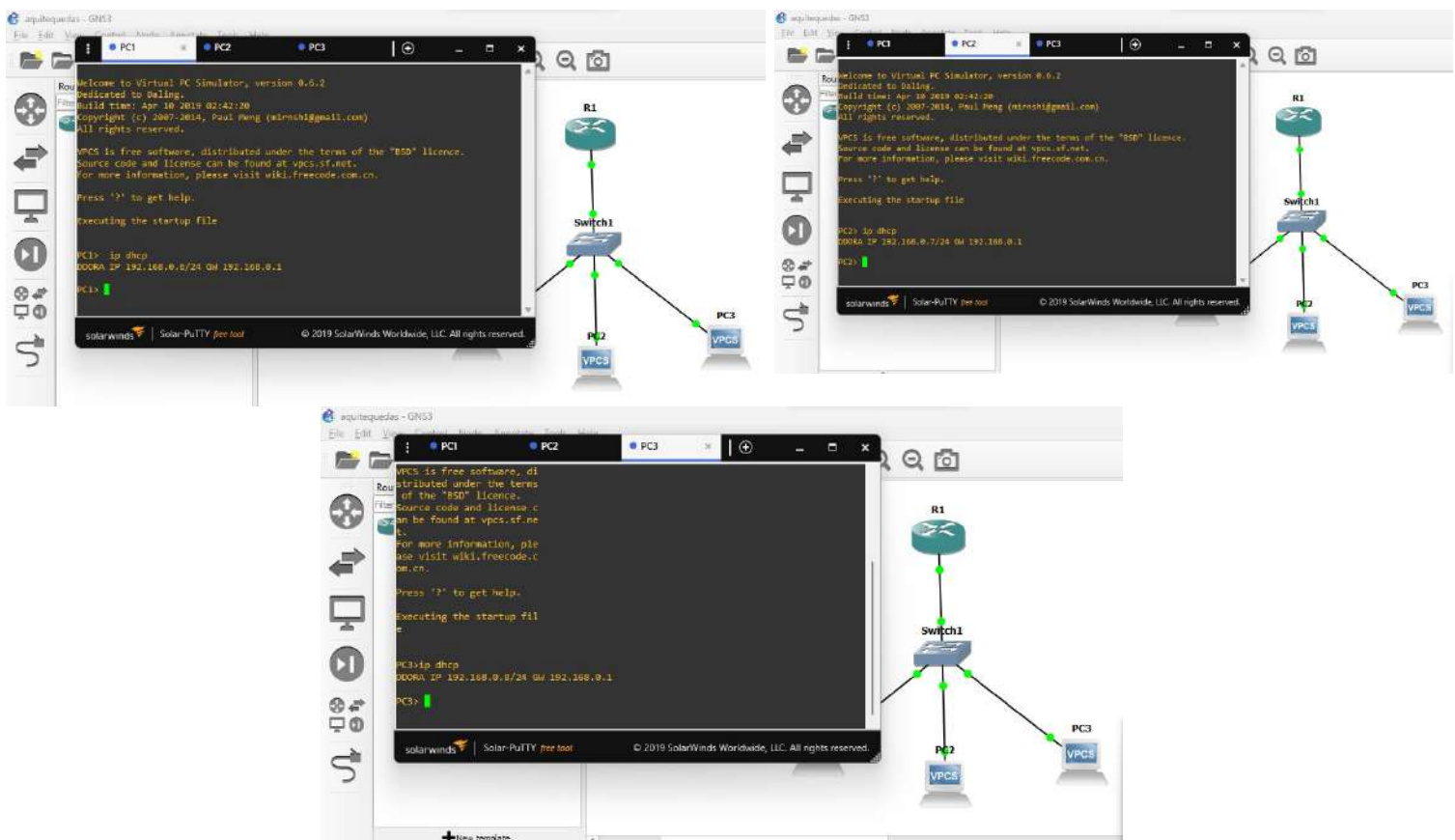
Configuración del Router:

The screenshot shows the GNS3 interface with the console window open for router R1. The console output shows the configuration process, including entering configuration mode, setting the interface, and applying the configuration.

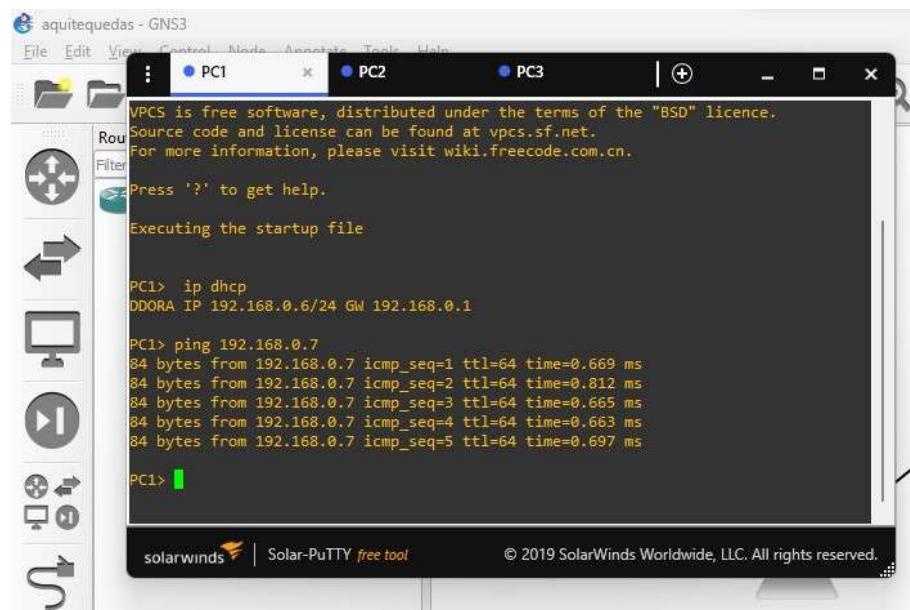
```
Mar 1 00:00:07.459: %LINK-5-CHANGED: Interface Serial12, changed state to admin
Administratively down
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int fa0/0
^
% Invalid input detected at '^' marker.
R1(config)#int fa0/0
^
% Invalid input detected at '^' marker.
R1(config)#int fa
R1(config)#int fastEthernet 0/0
^
% Invalid input detected at '^' marker.
R1(config)#exit
R1#ex
Mar 1 00:04:50.427: %SYS-5-CONFIG_I: Configured from console by console
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int
R1(config)#interface fa 0/0
^
% Invalid input detected at '^' marker.
R1(config)#
R1(config)#inter
R1(config)#interface fa
R1(config)#interface fastEthernet 0/0
^
% Invalid input detected at '^' marker.
R1(config)#interface fastEthernet 0
R1(config-if)#ip address 192.168.0.1 255.255.255.0
R1(config-if)#no sh
R1(config-if)#
```

Mostrando ip DHCP en las PCs:



Haciendo Ping entre PC1 y PC2:

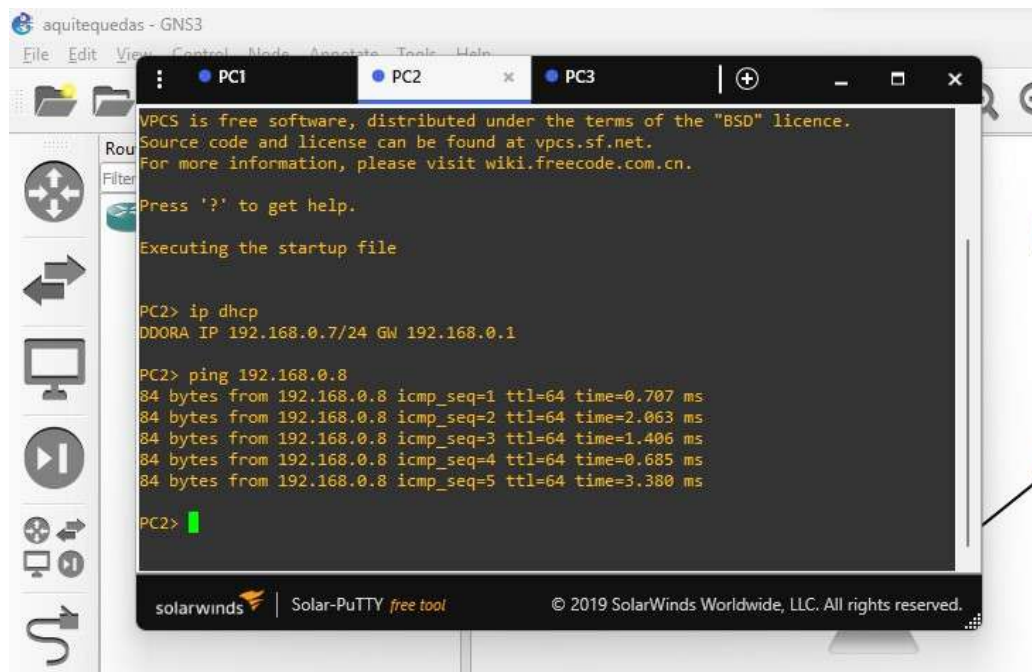


The screenshot shows a SolarWinds Solar-PuTTY terminal window titled 'aquitequedas - GNS3'. The window has tabs for PC1, PC2, and PC3. The PC1 tab is active, displaying the following text:

```
VPCS is free software, distributed under the terms of the "BSD" licence.  
Source code and license can be found at vpcs.sf.net.  
For more information, please visit wiki.freecode.com.cn.  
  
Press '?' to get help.  
  
Executing the startup file  
  
PC1> ip dhcp  
DDORA IP 192.168.0.6/24 GW 192.168.0.1  
  
PC1> ping 192.168.0.7  
84 bytes from 192.168.0.7 icmp_seq=1 ttl=64 time=0.669 ms  
84 bytes from 192.168.0.7 icmp_seq=2 ttl=64 time=0.812 ms  
84 bytes from 192.168.0.7 icmp_seq=3 ttl=64 time=0.665 ms  
84 bytes from 192.168.0.7 icmp_seq=4 ttl=64 time=0.663 ms  
84 bytes from 192.168.0.7 icmp_seq=5 ttl=64 time=0.697 ms  
  
PC1> █
```

The bottom of the window shows the SolarWinds logo, 'Solar-PuTTY free tool', and the copyright notice '© 2019 SolarWinds Worldwide, LLC. All rights reserved.'

Haciendo Ping entre PC2 y PC3:



The screenshot shows the same SolarWinds Solar-PuTTY terminal window, but now the PC2 tab is active. The PC1 tab is still visible in the background. The PC2 tab displays the following text:

```
VPCS is free software, distributed under the terms of the "BSD" licence.  
Source code and license can be found at vpcs.sf.net.  
For more information, please visit wiki.freecode.com.cn.  
  
Press '?' to get help.  
  
Executing the startup file  
  
PC2> ip dhcp  
DDORA IP 192.168.0.7/24 GW 192.168.0.1  
  
PC2> ping 192.168.0.8  
84 bytes from 192.168.0.8 icmp_seq=1 ttl=64 time=0.707 ms  
84 bytes from 192.168.0.8 icmp_seq=2 ttl=64 time=2.063 ms  
84 bytes from 192.168.0.8 icmp_seq=3 ttl=64 time=1.406 ms  
84 bytes from 192.168.0.8 icmp_seq=4 ttl=64 time=0.685 ms  
84 bytes from 192.168.0.8 icmp_seq=5 ttl=64 time=3.380 ms  
  
PC2> █
```

The bottom of the window shows the SolarWinds logo, 'Solar-PuTTY free tool', and the copyright notice '© 2019 SolarWinds Worldwide, LLC. All rights reserved.'