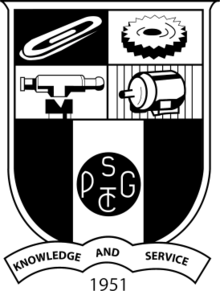
**RED HAT PACKAGE MANAGER (RPM)**

**A REPORT ON PACKAGE**

**Subject:** Operating System

****

DEPARTMENT OF APPLIED MATHEMATICS AND

COMPUTATIONAL SCIENCES, PSG COLLEGE OF

TECHNOLOGY, COIMBATORE-641004

**SUBMITTED BY**

Narendran K , 18PT24

Prem Kumar G,18PT26

**TABLE OF CONTENTS**

**ABSTRACT**

1.1 Introduction

1.2 Description

1.3 Tools & Technology

1.4 Work flow

1.5 Conclusion

1.6 Bibliography

**Abstract**:

Through this project we will attempt to learn and highlight the basic yet essential functionalities of the red hat package manager. These mainly include building, verifying and installing a rpm package. For this purpose, we will beusing a simple Tetris game built in C#.

The concept of RPM packaging can be overwhelming for first-timers because of the impression a steep learning curve is involved. In this article, we will demonstrate that building an RPM with minimal knowledge and experience is possible.

**Introduction: -**

The Red Hat Package Manager, commonly known as RPM, is a default open source and most popular package management utility for Red Hat based systems like RHEL, CentOS, Fedora.

The utility works only with packages built for processing by the rpm package.An RPM package is not just a pile of files. It’s also a pile of metadata to help with the management of those files.Installing, uninstalling, and upgrading RPM packages can be accomplished with short commands. RPM maintains a database of installed packages and their files, so you can invoke powerful queries and verifications on your system.

One of the main reasons to use rpm over zip files is that during upgrades, RPM handles configuration files carefully, so that you never lose your customizations. Something that you cannot accomplish with regular zip files.

For the developer, RPM allows you to take software source code and package it into source and binary packages for end users. This process is quite simple and is driven from a single file and optional patches that you create.

An RPM package can be signed using Gnu Privacy Guard key (or GnuPG), to help you make certain your downloaded package is trustworthy.During installation, GnuPG is installed by default. That way you can immediately start using GnuPG to verify any packages that you receive from Red Hat. Before doing so, you must first import Red Hat's public key.

**Description**

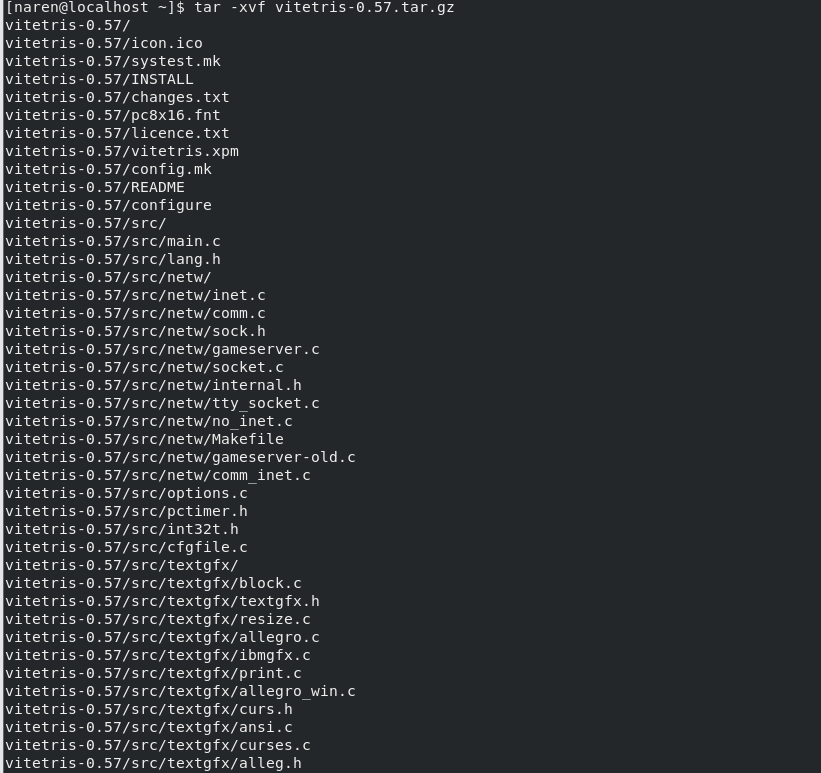
Tetris is a title-matching puzzle video game originally designed and programmed by Soviet Russian Software engineer Alexey Pajitnov in 1984 for the Electronika 60.

*Tetris* challenges the player to create complete lines by moving differently-shaped pieces – the tetrominoes, which scroll from the top to bottom of the playing field. The completed lines disappear and grant the player points, and the player can proceed to fill the vacated spaces. The game ends when the playing field is filled to the point that additional pieces can no longer descend. The longer the player can delay this inevitable outcome, the higher their score will be. In multiplayer games, the players must last longer than their opponents, and in certain versions, players can inflict penalties on opponents by completing a significant number of lines. Some adaptations have provided variations to the game's theme, such as three-dimensional displays or a system for reserving pieces.

**Tools and Technology**

To run this package or red hat package commands in general we require Red Hat Enterprise Linux, CentOS 7/8 or Fedora. For this we have used CentOS 7/8 as it is simple to use and it comes with rpm and yum pre-installed. We have mainly used these two package managers to create a rpm package.To install your development environment on a[Red Hat Enterprise Linux](https://developers.redhat.com/topics/linux/) (RHEL) system, we also require the following repositories to build RPMs: rhel-7-server-rpms, rhel-7-server-extras-rpms, and rhel-7-server-optional-rpms.

**Work flow**

1. We need to download the zip file present in the repository.
2. Use the tar command to unzip this .tar.gz file.
3. cat Makefile#INSTALL = install -oroot -groot # non-root users building the rpm won't be able to set this and the RPM build will fail.



1. Install rpm and all its dependencies



5. As a general rule, open source software written in C is compiled with three steps: configure, make, and make install. However, it is worth just running make on its own to see whether the software will build.

As a non-root user, test to see whether running make is sufficient:



6. To test the game, simply run ./tetris to see whether the game loads and can be played:



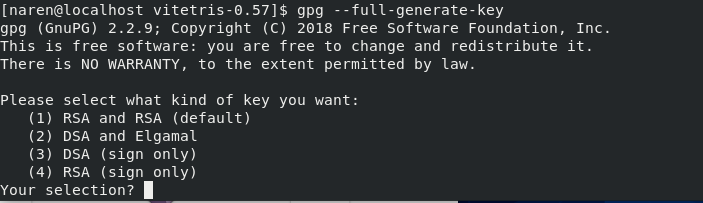
You will get the following output:



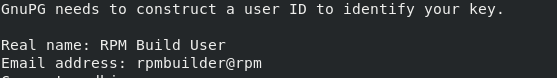
7.Signing your RPM package adds an extra layer of trust to your packages. Therefore, to create GPG keys as a non-root user, you will need a session with root access to run rngd.



8. To generate a GPG key you must switch to a non root user. Then run the following command:



Choose desired encryption method and enter following details when prompted.



You will be required to create a passphrase here.This passphrase will be used in the future when you are signing your rpm package.

9.To build the RPM, first create the build tree and the spec file.



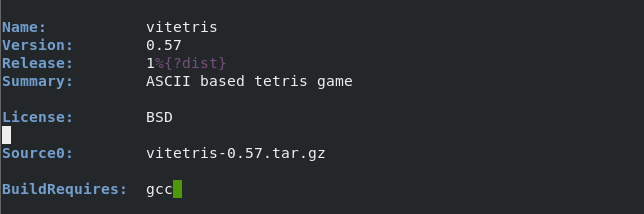
10.Place source code in the correct place.



11. Locate the spec file.

### 

Now provide the initial details about the package.



### 12.Test the initial build with the information just provided in the spec file.



### 13.Set %\_gpg\_name with the public key value.

### 

### 

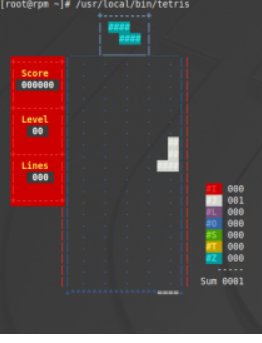
### 14.Sign the RPM package.



### 15.Test an installof the RPM as root user.



You should get the following screen:



## **Conclusion**

Depending on the software you intend to package, custom RPM packaging can be challenging. In this project, we aimed to do as much as possible to demonstrate the settings that are often required to build an RPM package from scratch.

**Bibliography**:

The following web references have been used while working on this project.

1. <http://isoredirect.centos.org/centos/8/isos/x86_64/CentOS-8.1.1911-x86_64-dvd1.iso>
2. <https://opensource.com/article/18/9/how-build-rpm-packages>
3. <https://docs.fedoraproject.org/ro/Fedora_Draft_Documentation/0.1/html/RPM_Guide/ch-command-reference.html>
4. <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-rpm>
5. <https://en.wikipedia.org/wiki/Tetris>