

Introducción a gmsh

Parte II: bash, g++, cmake, gnuplot, gmsh, paraview



Practique los ejemplos en Gitpod



Open in Gitpod

Disponible en



¡Únete al grupo en Telegram!



Acerca de gmsh

Es un software de código abierto para mallas tridimensionales de elementos finitos y visualización de datos, cuenta su propio lenguaje de scripting y cuatro módulos:

geometry capa geométrica orientado a objetos para la creación de formas tridimensionales.

Motores disponibles: Gmsh CAD y OpenCascade.

meshing un módulo de mallado

solving un módulo solucionador de ecuaciones

post-processing un módulo de postprocesador.

Estamos interesados en usar la interfaz de línea de comandos gmsh, además de conocer dicho lenguaje de scripting así como sus APIs.

Un archivo.geo $\xrightarrow{\text{gmsh}}$ archivo.msh. El primero contiene información de la geometría y parámetros de la malla.

Si estamos usando el repositorio arch4edu o el aur, instale con `yay -Sy gmsh gmsh-docs`.

```
Repository      : arch4edu
Name            : gmsh
Version         : 4.9.5-2
Description     : An automatic 3D finite element mesh generator with pre and post-processing facilities.
Architecture    : x86_64
URL             : http://gmsh.info/
Licenses        : custom
Groups          : None
Provides        : None
Depends On      : fltk med opencascade cairo metis alglib ann glu cgns lapack
Optional Deps   : gmsh-docs: docs for gmsh
                  python2: for gmsh.py
                  python:  for gmsh.py
                  julia:   for gmsh.jl
Conflicts With  : None
Replaces        : None
Download Size   : 8.81 MiB
Installed Size  : 39.68 MiB
Packager        : calil (on behalf of Jingbei Li) <i@jingbei.li>
Build Date     : Tue 08 Mar 2022 02:18:24 AM -05
Validated By    : MD5 Sum SHA-256 Sum Signature
```

Comandos elementales

Objetivos de esta introducción

A number of advanced.

- ▶ Correr las imágenes prediseñadas con el programa `docker` en sistemas tipo Unix.
- ▶ Crear una malla con `gmsh` a través de la API (C/C++, Python, Julia).
- ▶ Realizar los ejemplos resueltos del DUNE BOOK.

Conociendo el sistema operativo Arch Linux

Es una distribución GNU/Linux de propósito general, desarrollada¹ independientemente para procesadores x86-64, que se adhiere a los principios de simplicidad, modernidad, pragmatismo, centrado en usuarios y versatilidad.

Proporciona las últimas versiones estables de la mayoría del software siguiendo el modelo de lanzamiento continuo, no existen versiones como en Ubuntu 20.04, 21.10, 22.04, etc.

Arch está respaldado por pacman, un gestor de paquetes ligero, sencillo y rápido, que permite actualizar todo el sistema con una orden. Los scripts PKGBUILD aportados por la comunidad para la elaboración desde las fuentes, como los módulos de DUNE, se encuentran en el *Arch User Repository*.

Cuadro: Comparación de la línea de comando del gestión de software (fuente: wiki.archlinux.org)

Acción	Arch	Red Hat/Fedora	Debian/Ubuntu	SLES/openSUSE	Gentoo
Instala paquetes	<code>pacman -S</code>	<code>dnf install</code>	<code>apt install</code>	<code>zypper install</code>	<code>emerge -a</code>
Elimina paquetes	<code>pacman -Rs</code>	<code>dnf remove</code>	<code>apt remove</code>	<code>zypper remove</code>	<code>emerge -C</code>
Busca paquetes	<code>pacman -Ss</code>	<code>dnf search</code>	<code>apt search</code>	<code>zypper search</code>	<code>emerge -S</code>
Actualiza paquetes	<code>pacman -Syu</code>	<code>dnf upgrade</code>	<code>apt update && apt upgrade</code>	<code>zypper update</code>	<code>emerge -u world</code>

¹El líder del proyecto es Anthraxx, desarrollador alemán del kernel linux-hardened.

En esta ocasión hemos elegido Arch Linux como ambiente de trabajo porque tiene disponible una gran variedad de módulos de DUNE. Es recomendable habilitar el repositorio arch4edu².

²Administrado por Jingbei Li de la Universidad de Tsinghua.

```
gitpod ~/dune-basics $ cd
```




Listado 1: Programa hello-linux.cc.

```
// Tomado de https://stackoverflow.com/a/66161001

#include <sys/utsname.h>
#include <iostream>

// un pequeño ayudante para mostrar el contenido de una estructura utsname:
std::ostream &operator<<(std::ostream &os, const utsname &u)
{
    return os << "sysname : " << u.sysname << '\n'
        << "nodename: " << u.nodename << '\n'
        << "release : " << u.release << '\n'
        << "version : " << u.version << '\n'
        << "machine : " << u.machine << '\n';
}

int main()
{
    utsname result; // declarar la variable para contener el resultado

    uname(&result); // llamar a la función uname() para completar la estructura

    std::cout << result; // mostrar el resultado usando la función ayudante
}
```

Referencias

► Libros



Oliver Sander. *DUNE — The Distributed and Unified Numerics Environment*. First. Lecture Notes in Computational Science and Engineering 140. Springer International Publishing, 2020. ISBN: 978-3-030-59701-6. DOI: 10.1007/978-3-319-03038-8.

► Artículos



Peter Bastian y col. "The Dune framework: Basic concepts and recent developments". En: *Computers & Mathematics with Applications* 81.1 (1 de ene. de 2021). Development and Application of Open-source Software for Problems with Numerical PDEs, págs. 75-112. ISSN: 0898-1221. DOI: <https://doi.org/10.1016/j.camwa.2020.06.007>. URL: <https://www.sciencedirect.com/science/article/pii/S089812212030256X>.

► Sitios web



Santiago Torres Arias, Jesús Castro y Andrea Gómez. *Taller de contribución a Arch Linux – Cumbre de Contribuidores de Open Source Software (CCOSS)*. 22 de oct. de 2020. URL: <https://sg.com.mx/buzz/ponencias/ccoss-2020/taller-de-contribucion-arch-linux> (visitado 10-03-2021).



The Open Group. *The Evolution of the Unix Time-sharing System*. 15 de ago. de 2021. URL: https://unix.org/what_is_unix/history_timeline.html (visitado 30-05-2021).