Manuales de instalación

Luis Valenzuela

1. HIJING

HIJING (Heavy Ion Jet INteraction Generator) fue desarrollado inicialmente en Fortran 77 para simular la producción de partículas y jets múltiples en colisiones pp, pA, y AB a altas energías. Además proporciona condiciones iniciales para modelos de partones y cascadas de hadrones.

Primero se debe descomprimir el archivo $hijing1411.tgz$ [1], el cual contiene:

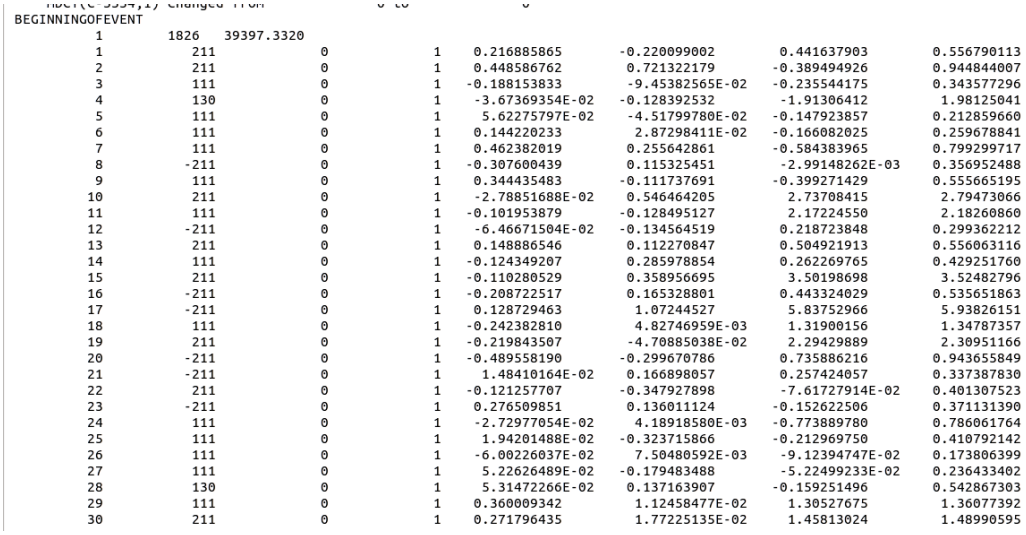
- hipyset1.35.f Contiene la subrutina de PYTHIA
- hijing1.411.f Contiene la subrutina de HIJING y llama a PYTHIA
- test.f Lee los datos de entrada e inicia HIJING. Contiene la semilla generadora de números aleatorios, la energía de la colisión en el cms, las partículas colisionadoras o núcleos, número de eventos, e información sobre las partículas que decaen.
- test.in Datos de entrada
- Makefile Archivo para compilar

Hijing se puede compilar mediante el archivo Makefile usando el comando *make* en su respectivo directorio.

Una vez compilado se puede correr de la forma:

```
./test.exe < test.in > test.out
```

Como resultado genera el archivo *test.out* donde se enlistan los datos asociados a las trazas generadas en todos los eventos. Al inicio de cada evento se especifica la cantidad de partículas generadas, enseguida se enlista cada partícula, siendo etiquetada con el número que la identifica, los decaimientos, y su respectivo cuadrimomento. En la figura 1 se muestra una fracción de los datos de uno de los tantos eventos generados por HIJING para colisiones $Au + Au$ a $200GeV$ en el cms.



```
BEGINNINGOFEVENT
          1        1826   39397.3320
          1         211            0            1    0.216885865       -0.220099002        0.441637903        0.556790113
          2         211            0            1    0.448586762        0.721322179       -0.389494926        0.944844007
          3         111            0            1   -0.188153833       -9.45382565E-02   -0.235544175        0.343577296
          4         130            0            1    -3.67369354E-02   -0.128392532       -1.91306412         1.98125041
          5         111            0            1     5.62275797E-02    -4.51799780E-02   -0.147923857        0.212859660
          6         111            0            1    0.144220233        2.87298411E-02   -0.166082025        0.259678841
          7         111            0            1    0.462382019        0.255642861       -0.584383965        0.799299717
          8        -211            0            1   -0.307600439        0.115325451        -2.99148262E-03    0.356952488
          9         111            0            1    0.344435483       -0.111737691       -0.399271429        0.555665195
         10         211            0            1    -2.78851688E-02    0.546464205         2.73708415         2.79473066
         11         111            0            1   -0.101953879       -0.128495127         2.17224550         2.18260860
         12        -211            0            1    -6.46671504E-02   -0.134564519        0.218723848        0.299362212
         13         211            0            1    0.148886546        0.112270847        0.504921913        0.556063116
         14         111            0            1   -0.124349207        0.285978854        0.262269765        0.429251760
         15         211            0            1   -0.110280529        0.358956695         3.50198698         3.52482796
         16        -211            0            1   -0.208722517        0.165328801        0.443324029        0.535651863
         17        -211            0            1    0.128729463         1.07244527         5.83752966         5.93826151
         18         111            0            1   -0.242382810         4.82746959E-03     1.31900156         1.34787357
         19         211            0            1   -0.219843507        -4.70885038E-02     2.29429889         2.30951166
         20        -211            0            1   -0.489558190       -0.299670786        0.735886216        0.943655849
         21        -211            0            1     1.48410164E-02    0.166898057        0.257424057        0.337387830
         22         211            0            1   -0.121257707       -0.347927898        -7.61727914E-02    0.401307523
         23        -211            0            1    0.276509851        0.136011124       -0.152622506        0.371131390
         24         111            0            1    -2.72977054E-02     4.18918580E-03   -0.773889780        0.786061764
         25         111            0            1     1.94201488E-02   -0.323715866       -0.212969750        0.410792142
         26         111            0            1    -6.00226037E-02     7.50480592E-03    -9.12394747E-02    0.173806399
         27         111            0            1     5.22626489E-02   -0.179483488        -5.22499233E-02    0.236433402
         28         130            0            1     5.31472266E-02    0.137163907       -0.159251496        0.542867303
         29         111            0            1    0.360009342         1.12458477E-02     1.30527675         1.36077392
         30         211            0            1    0.271796435         1.77225135E-02     1.45813024         1.48990595
```

Figura 1: Datos de salida de HIJING.

2. MpdRoot

MpdRoot requiere de una serie de paqueterías antes de su instalación.

2.1. Paqueterías

- Se necesitan paqueterías de acuerdo al tipo del sistema operativo, en el caso del ambiente Debian, como Ubuntu, las paqueterías se pueden instalar de la siguiente forma:

```
sudo apt-get install subversion git make cmake gcc gfortran binutils patch libx11-dev
libxmu-dev libxpm-dev libxft-dev libxext-dev dpkg-dev xlibmesa-glu-dev libglew-dev
libxml2-dev libexpat1-dev zlib1g-dev libpqxx3-dev libmysqlclient-dev
libcurl4-openssl-dev automake libtool fftw3-dev
```

- También se necesita una serie de paqueterías externas, conocidas en conjunto como Fairsoft. Son instaladas usualmente en el directorio opt. Se descarga con las siguientes instrucciones:

```
cd /opt
git clone https://github.com/FairRootGroup/FairSoft.git fairsoft
cd fairsoft
git checkout may16p1
```

Para soporte de GCC:

```
wget http://mpd.jinr.ru/data/fairsoft_may16p1.patch
patch -p1 -i fairsoft_may16p1.patch
```

- Ahora se puede instalar Fairsoft mediante

```
./configure.sh
```

- Se sugiere hacer la instalación en el directorio $/opt/fairsoft/install$, y seleccionar la versión 5 de Root, pues root 6 no es compatible con MpdRoot.

2.2. MpdRoot

Una vez instaladas las paqueterías necesarias se puede realizar la instalación de MpdRoot. Se descarga MpdRoot:

```
git clone --recursive https://git.jinr.ru/nica/mpdroot.git
```

- En esta parte es común obtener el siguiente error:

```
fatal: unable to access 'https://git.jinr.ru/nica/mpdroot.git/':gnutls_handshake()
failed: Handshake failed
```

- Para solucionarlo se procede instalando build-essential, fakeroot y dpkg-dev mediante el comando:

```
sudo apt-get install build-essential fakeroot dpkg-dev
```

- Se crea un directorio llamado git-rectify en el directorio home, se accede a él y se descarga el git:

```
mkdir ~/git-rectify
cd ~/git-rectify
apt-get source git
```

- Se instalan las dependencias de git:

```
sudo apt-get build-dep git
```

- Se instalan las librerías libcurl:

```
sudo apt-get install libcurl4-openssl-dev
```

- Para desempacar los paquetes se usa el comando:

```
dpkg-source -x git_1.9.1-1ubuntu0.1
```

Donde el nombre `git_1.9.1-1ubuntu0.1` puede variar dependiendo de la versión de Ubuntu.

- Se accede al directorio `git_ 1.9.1` y se abre el archivo `git_1.9.1/debian/control` con un editor de texto. Enseguida, en el archivo se reemplaza $libcurl4 - gnutls - dev$ por $libcurl4 - openssl - dev$. También se abre con un editor de texto el archivo $debian/rules$ y se elimina la línea $TEST = test$.

- Se construye la paquetería mediante el comando:

```
sudo dpkg-buildpackage -rfakeroot -b
```

- Finalmente se instala la paquetería mediante la siguiente instrucción, notando que el nombre del git varía dependiendo de la arquitectura del sistema:

```
sudo dpkg -i git_1.7.9.5-1_amd64.deb
```

Ahora debe ser posible descargar MpdRoot mediante el git.

- Entonces, regresando al paso anterior:

```
git clone --recursive git@git.jinr.ru:nica/mpdroot.git
```

- Accedemos al directorio mpdroot y creamos un directorio build. Entonces construimos y compilamos MpdRoot de la siguiente forma:

```
cd mpdroot
mkdir build
. SetEnv.sh
cd build
cmake ..
```

- Para construir el sistema:

```
make
. config.sh
```

- Aquí es importante notar que se debe correr . config.sh cada vez que se vaya a trabajar con MpdRoot.

- Finalmente se actualiza:

```
cd mpdroot
git pull --recurse-submodules
```

3. FairRoot

- FairRoot también necesita de Fairsoft como pre-requisito de instalación. Después de instalado debe exportarse la variable SIMPATH en el directorio de instalación de FairSoft. Siguiendo el directorio de instalación predeterminado es de la forma:

```
export SIMPATH=/opt/fairsoft/install
```

o de la forma:

```
export PATH={SIMPATH}: /opt/fairsoft/install
```

- Ahora se descarga FairRoot:

```
git clone https://github.com/FairRootGroup/FairRoot.git
```

Si se obtiene el problema Gnutls handshake Failed GIT se puede proceder como en el caso anterior.

- Ahora creamos un directorio para construir FairRoot, y accedemos a él:

```
mkdir build
cd build
```

- Se crean los archivos makefile para compilar FairRoot:

```
cmake -DCMAKE_INSTALL_PREFIX="Aquí se especifica el directorio de instalación"
                             "Aquí va el directorio donde se descargó FairRoot"
```

- En mi caso particular es de la forma:

```
cmake -DCMAKE_INSTALL_PREFIX= /home/luisval/build_FairRoot /home/luisval/FairRoot
```

- Finalmente compilamos mediante el comando make:

```
make
```

4. AliRoot

Hay varias formas de instalar AliRoot, de la que hablaré a continuación es la forma manual antigua. Esta instalación se compone de ROOT, GEANT3 y AliRoot Core, pues para mi interés, que es trabajar con uno de los generadores que contiene AliRoot, HIJING, es suficiente.

4.1. Pre-requisitos

- Como pre-requisito es necesario desinstalar ROOT. Para verificar si hay alguna versión de ROOT instalada (de haber instalado FairSoft sí la hay), se emplea el comando:

```
dpkg --get-selections | grep root-system
```

- Si aparece alguna instalación de ROOT se desinstala:

```
sudo apt-get purge root-system-XXX root-system-YYY...
```

Nota: No es estrictamente necesario desinstalar ROOT. Se puede tener el cuidado de evitar problemas eliminando *sourcebin/thisroot.sh* o cambiando la forma de iniciar el ambiente ROOT.

- Para instalar las paqueterías necesarias primero actualizamos las paqueterías disponibles:

```
sudo apt-get update
```

- Las instalamos:

```
sudo apt-get install curl build-essential gfortran subversion
cmake libmysqlclient-dev xorg-dev libglu1-mesa-dev libfftw3-dev
libssl-dev libxml2-dev libtool automake git unzip libcgal-dev
```

- Se sugiere instalar Clang para compilar:

```
sudo apt-get install clang-3.4
```

- Es importante tener al menos la versión CMake v2.8.12. Entonces verificamos con el comando:

```
 cmake --version
```

Si no se encuentra procedemos con su instalación.

- Ahora verificamos si se encuentra git-new-workdir:

```
which git-new-workdir
```

- Si no se encuentra se puede instalar a través de los siguientes comandos:

```
sudo curl -L https://raw.github.com/gerrywastaken/git-new-workdir/master/git-new-workdir
-o /usr/bin/git-new-workdir sudo chmod 0777 /usr/bin/git-new-workdir
```

4.2. Instalación de AliRoot

- Creamos un directorio donde estará todo el software de ALICE:

```
mkdir $HOME/alicesw
```

- Accedemos al directorio y descargamos $alice - env.sh$:

```
cd $HOME/alicesw
curl -L https://raw.github.com/dberzano/cern-alice-setup/master/alice-env.sh -o alice-env.sh
```

- Enseguida se ejecuta el siguiente comando para generar el archivo $alice - env.conf$:

```
source alice-env.sh
```

- Accedemos al archivo bash $/.bashrc$ y agregamos la siguiente línea para cargar las variables del ambiente ali:

```
alias ali='source $HOME/alicesw/alice-env.sh'
```

- Ahora para cargar el ambiente de AliRoot basta con escribir ali en la terminal.

4.3. ROOT

- Creamos un directorio para la instalación de ROOT:

```
mkdir -p "$ALICE_PREFIX/root/git"
```

- Accedemos al directorio y descargamos ROOT:

```
cd "$ALICE_PREFIX/root/git"
git clone https://github.com/alisw/root.git .
```

- Se carga el ambiente de AliRoot mediante el comando *ali*, y se especifica la versión de ROOT que se desea instalar y se usan los siguientes comandos para actualizar:

```
cd "$ALICE_PREFIX/root/git"
git remote update --prune origin
```

Ahora usamos el siguiente comando para limpiar rastros de cualquier versión de ROOT instalada anteriormente.

```
[[ -f "$ALICE_PREFIX/root/$ROOT_SUBDIR/LICENSE" ]] && rm -rf "$ALICE_PREFIX/root/$ROOT_SUBDIR"
```

Creamos un directorio git para la versión deseada de ROOT:

```
mkdir -p "$ALICE_PREFIX/root/$ROOT_SUBDIR"
git-new-workdir "$ALICE_PREFIX/root/git/" "$ALICE_PREFIX/root/$ROOT_SUBDIR/src/" "$ROOT_VER"
```

- Se crea un directorio para construir ROOT en él y se accede a dicho directorio:

```
mkdir -p "$ALICE_PREFIX/root/$ROOT_SUBDIR/build/"
cd "$ALICE_PREFIX/root/$ROOT_SUBDIR/build/
```

- Procediendo, configuramos la compilación de ROOT. Para compilar con GCC se usan las siguientes líneas:

```
"$ALICE_PREFIX/root/$ROOT_SUBDIR/src/configure" \
  --with-pythia6-uscore=SINGLE \
  --with-monalisa-incdir="$GSHELL_ROOT/include" \
  --with-monalisa-libdir="$GSHELL_ROOT/lib" \
  --with-xrootd=$GSHELL_ROOT \
  --enable-minuit2 \
  --enable-roofit \
  --enable-soversion \
  --disable-bonjour \
  --enable-builtin-ftgl \
  --enable-builtin-freetype \
  --with-f77=$( which gfortran ) \
  --with-cc=$( which gcc ) \
  --with-cxx=$( which g++ ) \
  --with-ld=$( which g++ ) \
  --prefix="$ROOTSYS" \
  --incdir="$ROOTSYS/include" \
  --libdir="$ROOTSYS/lib" \
  --datadir="$ROOTSYS" \
  --etcdir="$ROOTSYS/etc"
```

- Para compilar con clang

```
"$ALICE_PREFIX/root/$ROOT_SUBDIR/src/configure" \
  --with-pythia6-uscore=SINGLE \
  --with-monalisa-incdir="$GSHELL_ROOT/include" \
  --with-monalisa-libdir="$GSHELL_ROOT/lib" \
  --with-xrootd=$GSHELL_ROOT \
  --enable-minuit2 \
  --enable-roofit \
  --enable-soversion \
  --disable-bonjour \
  --enable-builtin-ftgl \
  --enable-builtin-freetype \
  --with-clang \
  --with-f77=$( which gfortran ) \
  --with-cc=$( which clang ) \
  --with-cxx=$( which clang++ ) \
  --with-ld=$( which clang++ ) \
  --prefix="$ROOTSYS" \
  --incdir="$ROOTSYS/include" \
  --libdir="$ROOTSYS/lib" \
  --datadir="$ROOTSYS" \
  --etcdir="$ROOTSYS/etc"
```

Nota Se sugiere usar la opción de clang como compilador.

- Podemos compilar ROOT mediante el comando *make*:

```
make -j$MJ OPT='-O2 -g'
```

Ahora ya tenemos ROOT instalado. Para verificar si la instalación se hizo correctamente podemos proceder abriendo otra terminal, cargamos el ambiente de AliRoot mediante el comando ali, enseguida presionamos enter, cargamos el ambiente eligiendo la primera opción, y si ROOT aparece disponible, como en la figura 2, entonces la instalación está bien. A partir de aquí, para la instalación de cualquier software en AliRoot debe primero cargarse el ambiente de AliRoot de la misma forma.

Figura 2: Ambiente de AliRoot

4.4. GEANT3

- Primero cargamos el ambiente de AliRoot, como se explicó al final de la subsección anterior. Se crea un directorio para descargar GEANT3 y se accede a él:

```
 mkdir -p "$ALICE_PREFIX/geant3/git"
cd "$ALICE_PREFIX/geant3/git"
git clone http://root.cern.ch/git/geant3.git .
```

- Se actualiza el git:

```
cd "$ALICE_PREFIX/geant3/git"
git remote update --prune
```

- Limpiamos rastros de GEANT3 de instalaciones previas:

```
[[ -f "$ALICE_PREFIX/geant3/$G3_SUBDIR/README" ]] && rm -rf "$ALICE_PREFIX/geant3/$G3_SUBDIR"
```

- Se verifica la versión deseada con el siguiente comando:

```
git-new-workdir "$ALICE_PREFIX/geant3/git" "$ALICE_PREFIX/geant3/$G3_SUBDIR/src" "$G3_VER"
```

- Creamos el directorio donde se va a construir GEANT3:

```
mkdir -p "$ALICE_PREFIX/geant3/$G3_SUBDIR/build"
```

- Accedemos y corremos CMake para generar los makeFiles:

```
cd "$ALICE_PREFIX/geant3/$G3_SUBDIR/build/"
cmake "$ALICE_PREFIX/geant3/$G3_SUBDIR/src/"
-DCMAKE_INSTALL_PREFIX="$ALICE_PREFIX/geant3/$G3_SUBDIR/inst/"
```

- Finalmente compilamos GEANT3 mediante el comando make:

```
make -j$MJ
```

- Verificamos si la instalación se hizo bien, para ello cargamos el ambiente de AliRoot y comprobamos si GEANT3 se encuentra tal y como en la figura 2.

4.5. AliRoot Core

- Iniciamos cargando el ambiente de AliRoot.

- Se crea un clon del git:

```
[ ! -d "$ALICE_PREFIX"/aliroot/git ] && git clone http://git.cern.ch/pub/AliRoot
"$ALICE_PREFIX"/aliroot/git
```

- Se actualiza el git:

```
cd "$ALICE_PREFIX"/aliroot/git
git remote update --prune origin
```

- Usamos git-new-workdir:

```
git-new-workdir "${ALICE_PREFIX}/aliroot/git" "$(dirname "$ALICE_ROOT")/src" "$ALICE_VER"
```

- Se descarga el git:

```
cd "$(dirname "$ALICE_ROOT")/src"
git remote set-url origin http://git.cern.ch/pub/AliRoot
```

- Se crea un directorio para construir AliRoot y se accede a dicho directorio:

```
mkdir -p "$(dirname "$ALICE_ROOT")/build"
cd "$(dirname "$ALICE_ROOT")/build"
```

- Configuramos AliRoot mediante:

```
cmake "$(dirname "$ALICE_ROOT")/src" \
  -DCMAKE_C_COMPILER=‘root-config --cc‘ \
  -DCMAKE_CXX_COMPILER=‘root-config --cxx‘ \
  -DCMAKE_Fortran_COMPILER=‘root-config --f77‘ \
  -DCMAKE_MODULE_LINKER_FLAGS=’-Wl,--no-as-needed’ \
  -DCMAKE_SHARED_LINKER_FLAGS=’-Wl,--no-as-needed’ \
  -DCMAKE_EXE_LINKER_FLAGS=’-Wl,--no-as-needed’ \
  -DCMAKE_INSTALL_PREFIX="$ALICE_ROOT" \
  -DROOTSYS="$ROOTSYS" \
  -DCMAKE_BUILD_TYPE=RELWITHDEBINFO \
```

- Finalmente compilamos AliRoot mediante el comando *make*:

```
make -j$MJ
```

- Para verificar si la instalación se realizó bien, se carga el ambiente de AliRoot en otra terminal y se comprueba si AliRoot Core se encuentra en las opciones como en la figura 2.

Referencias

[1] HIJING1.411 recuperado en enero de 2017 de `http://atomfizika.elte.hu/haladolabor/HEP/instructions.html`

[2] How to install MpdRoot and BmnRoot `http://mpd.jinr.ru/howto-install-mpdroot/`

[3] Gnutls handshake Failed GIT `http://devopscube.com/gnutls-handshake-failed-aws-codecommit/`

[4] Export SIMPATH `https://git.jinr.ru/nica/bmnroot/blob/aec4f076fb8d14a3e2b33dbe84a475e863ea82a5/SetEnv.sh#L4`

[5] CMake`https://cmake.org/`