**Sniper 7.4**

* Sites:
  + <https://snipersim.org/w/The_Sniper_Multi-Core_Simulator>
* Preparação do ambiente:
  + Instruções de instalação:
    - <https://snipersim.org/w/Getting_Started>

* + Download:
    - wget <http://snipersim.org/download/dfbc471f39ee1a74/packages/sniper-latest.tgz>
  + Compilar Sniper
    - make
    - sudo apt-get update
    - sudo apt-get install libbz2-dev
    - sudo apt-get install bzip2-devel
    - sudo apt-get install libboost-dev
    - sudo apt-get install boost-devel
    - sudo apt-get install libsqlite3-dev
    - sudo apt-get install sqlite-devel
    - sudo apt-get install xsltproc libxmu-dev

* + Execução de programas
    - Acessar cada pasta de programas de teste e executar os comandos abaixo
      * make run > sniper-result-api.txt
      * make run > sniper-result-api.txt
      * make run > sniper-result-dvfs.txt
      * make run > sniper-result-fft.txt
      * make run > sniper-result-fft-dvfs.txt
      * make run > sniper-result-fft-hetero.txt
      * make run > sniper-result-fft-hetero-cfg.txt
      * make run > sniper-result-fft-marker.txt
      * make run > sniper-result-fork.txt
      * make run > sniper-result-shared.txt
      * make run > sniper-result-signal.txt
      * make run > sniper-result-smc.txt
      * make run > sniper-result-sniper-in-sniper.txt
      * make run > sniper-result-spinloop.txt
      * make run > sniper-result-true.txt
      * ../../run-sniper ./RADIX > sniper-result-radix.txt
      * ../../run-sniper ./CHOLESKY tk14.0 > sniper-result-cholesky.txt

* + Extras Benchmarks: <http://snipersim.org/packages/sniper-benchmarks.tbz>
    - Usar barnes
    - Benchmarks > splash2 > splash2 > codes 🡪 tem vários kernel para avaliação
      * Programa Barnes
      * Programa RADIX
* Preparação do ambiente:
  + Instruções de instalação:
    - <https://snipersim.org/w/Getting_Started>

* + Download:
    - wget <http://snipersim.org/download/dfbc471f39ee1a74/packages/sniper-latest.tgz>
  + Instalação:
    - printenv – mostra variáveis de ambiente
  + Remoção de versões mais recentes python
    - apt-get remove python3.11
    - apt-get remove python3.10
  + Instalação do python versão 2.7
    - apt install python2 python2-dev
    - pasta para links simbólicos: /bin
    - criar link simbólico de “python” para “python2.7”
      * ln -s python2.7 python
  + Acessar pasta do Pin
    - Criar link simbólico para a pasta Pin usando *ln -s <pasta-do-pin> ~/pin-kit* na pasta /bin.
      * ln -s /usr/local/pin-3.27-98718-gbeaa5d51e-gcc-linux pin-kit
  + Compilação (build) 🡪 NÃO FICOU BOA
    - export SNIPER\_TARGET\_ARCH=ia32
    - export SNIPER\_TARGET\_ARCH
    - make clean
    - make –j 2
  + Compilação (build)
    - sudo dpkg --add-architecture i386
    - Instalação individual dos pacotes com sucesso:
      * sudo apt-get install binutils
      * sudo apt-get install build-essential
      * sudo apt-get install curl
      * sudo apt-get install git
      * sudo apt-get install libboost-dev
      * sudo apt-get install libbz2-dev
      * sudo apt-get install libc6:i386
      * sudo apt-get install libncurses5:i386
      * sudo apt-get install libsqlite3-dev
      * sudo apt-get install libstdc++6:i386
      * sudo apt-get install python
      * sudo apt-get install wget
      * sudo apt-get install zlib1g-dev

* Execução do Sniper
  + run-sniper
  + acessar a pasta “test”
  + acessar a pasta de um programa específico
    - make
    - ./”nome\_programa”
  + Contar eventos de um programa
    - perf record ./fft
    - perf report
    - perf report –header

* Execução do sniper nos programas selecionados:
  + ./run-sniper programs\_selected/fft
* Programas Selecionados:
  + fft
* AAAAAAAAAAAAA
  + - setar variável de ambiente
      * export SNIPER\_TARGET\_ARCH=ia32
    - Indicar a versão do python 2.7 a ser utilizada
      * alias python=/usr/bin/python2.7
    - Compilar sniper:
      * make –j 2


    - sudo dpkg --add-architecture i386
    - sudo apt-get install binutils build-essential curl git libboost-dev libbz2-dev libc6:i386 libncurses5:i386 libsqlite3-dev libstdc++6:i386 python wget zlib1g-dev

    - sudo apt-get install binutils build-essential curl git libboost-dev libbz2-dev libc6:i386 libncurses5:i386 libsqlite3-dev libstdc++6:i386 python wget zlib1g-dev
    - 'make -j N' where N is the number of cores in your machine to use parallel make
      * make –j 2
    - ATENÇÂO: corrigi o erro no script checkdependency.py na instrução “print”, linha 49
      * Instalação dos seguintes pacotes:
        + apt-get install libboost-dev
        + apt-get install zlib1g-dev / zlib-devel
        + apt-get install libbz2-dev / bzip2-devel
        + apt-get install g++ / gcc-c++
        + apt-get install wget
        + apt-get install libsqllite3-dev
        + apt-get install libsqlite3-dev
      * Instalação dos pacotes python 2to3 para permitir processar comandos “print” sem parênteses
        + apt install 2to3
        + apt install python3-lib2to3
        + apt install python3-toolz
  + nv
* Realizar o build de todos os benchmarks:

Sugestão do Fabrício

* sudo dpkg --add-architecture i386
* sudo apt-get install binutils build-essential curl git libboost-dev libbz2-dev libc6:i386 libncurses5:i386 libsqlite3-dev libstdc++6:i386 python wget zlib1g-dev
* make