

Dezipar o arquivo .ova, que contém a imagem da máquina virtual. Para abrir o arquivo, pode-se utilizar qualquer programa compatível com o formato ([VirtualBox](#), por exemplo, é um programa gratuito e multiplataforma que pode ser utilizado). Uma vez que o programa (ex. VirtualBox) esteja instalado, importar o arquivo .ova disponibilizado.

A senha de acesso é *orgb*. Assim como o nome de usuário.

Comandos no terminal do linux indicados nesse tutorial são precedidos de '\$', exemplo:

```
$ comando_qualquer
```

xcDepois de logar no sistema linux pela máquina virtual, abrir o terminal, pela barra de aplicativos à esquerda, atalho na área de trabalho ou atalho Ctrl + Alt + T.

Para colar comandos exemplo no terminal utilize o atalho Ctrl + Shift + V. Se você quiser copiar e colar texto entre a sua máquina *host* (seu computador) e a máquina *guest* (sistema na máquina virtual) pode ser necessário instalar complementos da máquina virtual.

Acesse, via terminal, a pasta do gem5 com o comando Change Directory (cd):

```
$ cd gem5
```

Para listar os arquivos e diretórios, use o comando List (ls)

```
$ ls
```

Execute um programa exemplo (hello world) no gem5. Além do executável do gem5, é preciso especificar um arquivo de simulação python (simulate.py, dentro da pasta *orgb_configs*), a opção 'run-benchmark', a flag '-c' ou '--cmd' e o nome do programa. Se algum argumento precisar ser passado para o programa, ainda pode-se adicionar a flag '-o' seguido da lista de argumentos separados por espaços.

Por exemplo, o comando abaixo execute um hello world, já compilado, no gem5:

```
$ ./gem5 orgb_configs/simulate.py run-benchmark -c tests/test-progs/hello/bin/x86/linux/hello
```

Se a simulação estiver correta, um novo arquivo 'stats.txt' será gerado na pasta 'm5out', dentro da pasta do gem5. Você pode abrir o arquivo pela interface gráfica do Ubuntu, abrindo o navegador de pastas que também está na barra de aplicativos à esquerda.

Seus benchmarks devem ser compilados estaticamente para executarem no gem5. Uma pasta modelo para programas já está criada dentro da pasta do gem5. Acesse-a e compile o programa hello.c, exemplo:

```
$ cd orgb_progs
```

```
$ gcc hello.c -o hello -static
```

A flag `-o` indica o nome de saída do executável, e a flag `-static` indica que as bibliotecas necessárias serão estaticamente linkadas.

Volte para a pasta base do gem5 para testar o programa:

```
$ cd .
```

".." indica o diretório pai.

Simule no gem5 o arquivo `.c` que você acabou de compilar.

```
$ ./gem5 orgb_configs/simulate.py run-benchmark -c orgb_progs/hello
```

Um novo arquivo `stats.txt` estará disponível na pasta `m5out`.

O procedimento é semelhante para os benchmarks escolhidos para utilizar no trabalho. Compile e simule no gem5.

Finalmente, os parâmetros do processador ou cache podem ser modificados nos arquivos:

`orgb_configs/systems/cpus/MyO3CPU.py` (cpu)

e

`orgb_configs/systems/caches/basic_caches.py` (caches)