**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS**

**CENTRO DE CIÊNCIAS EXATAS, AMBIENTAIS E DE TECNOLOGIA**

**CHRISTOPHER OLIVEIRA RA:18726430**

**GIULIANO SANFINS RA:17142837**

**MATHEUS MORETTI RA:18082974**

**MURILO ARAUJO RA:17747775**

**VICTOR REIS RA:18726471**

**SISTEMAS OPERACIONAIS B - EXPERIMENTO 2 –**

**MÓDULOS DE KERNEL LINUX**

**CAMPINAS**

**2020**

**SUMÁRIO**

[**1.** **INTRODUÇÃO** 3](#_Toc50029931)

[**2.** **DISCUSSÃO** 3](#_Toc50029932)

[**3.** **CONCLUSÃO** 10](#_Toc50029933)

# **1. INTRODUÇÃO**

Este experimento consistiu no estudo mais aprofundado de um módulo de Kernel. Deste modo foi-nos dado conhecimento para a criação, compilação, instalação e remoção de um módulo, e assim criarmos módulos básicos, os famosos *“Hello World”,* para conhecer na prática como esta parte importante de um kernel realmente funciona. A habilidade adquirida neste experimento, será extremamente necessário para o caminhar do curso, onde será criado módulos mais complexos.

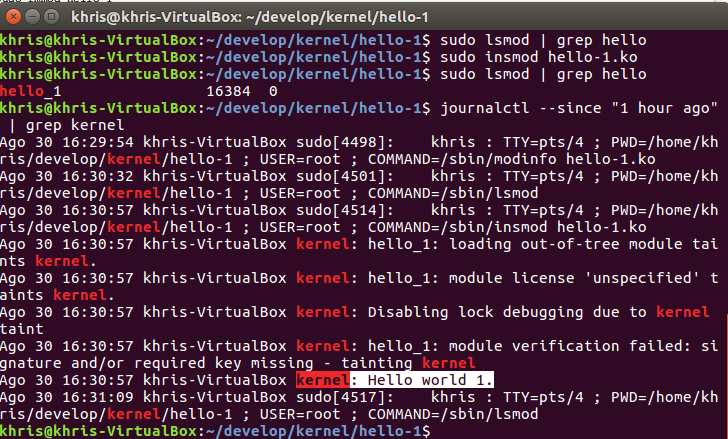
# **2. DISCUSSÃO**

Primeiramente instalamos o “pacote” *sudo apt-get install build-essential kmod*, no qual disponibiliza os comandos *modprobe, insmod* e *depmod*, que foram muito importantes para o desenvolvimento dessa atividade.

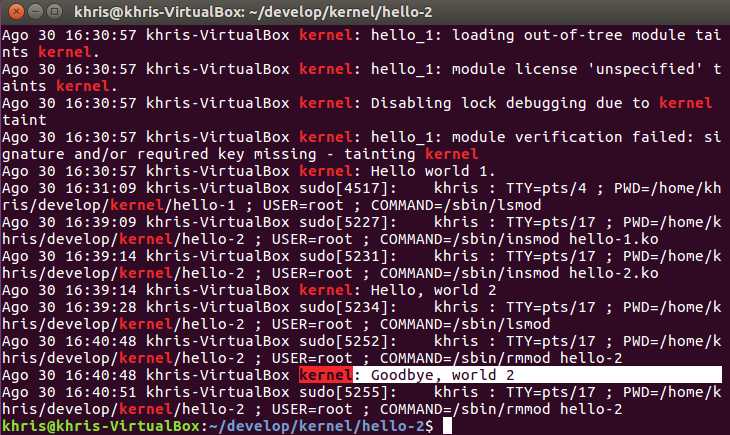
Para começar as execuções, precisávamos elaborar e salvar na mesma pasta do arquivo que vamos executar o arquivo *Makefile*. Logo em seguida realizamos o comando “*make”* para realizar a compilação do *Makefile* e a conversão do arquivo “.c” em “.ko”, como foi o que aconteceu nos arquivos abaixo

Utilizamos o comando *sudo rmmod* para realizar remoções (descarrega) dos arquivos que gostaríamos de remover. E o comando *journalctl --since "1 hour ago" |* grep kernel para visualizar todos os registros importantes da execução. E por último o comando *sudo lsmod* tem o principal objetivo de carregar os módulos que já estão carregados no kernel atual instalado na máquina.

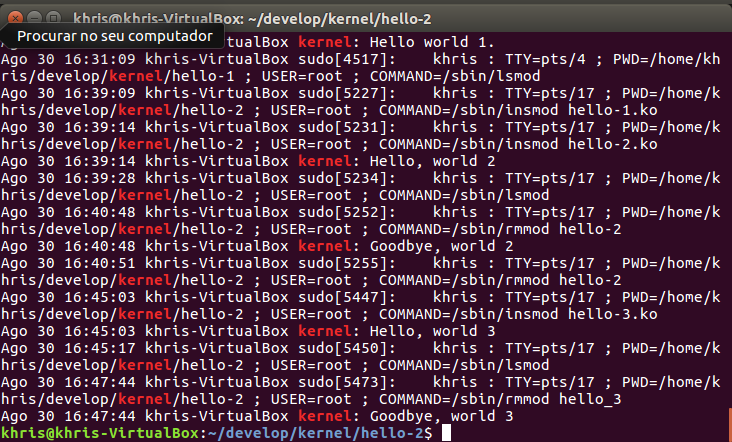
Desenvolvemos as atividades e todas as execuções dos programas, na ordem do *.pdf* disponibilizado pelo docente, abaixo pode se observar o resultado da execução do arquivo *“hello1-c”*:



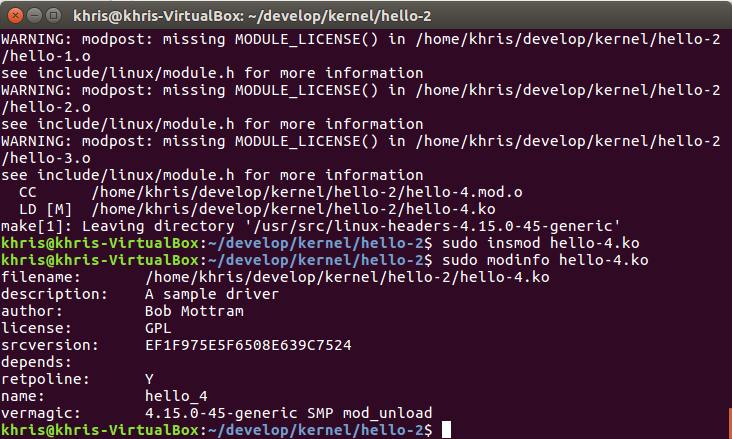
Abaixo podemos analisar o resultado da execução do arquivo *“hello2-c”,* com a realização dos mesmos procedimentos:



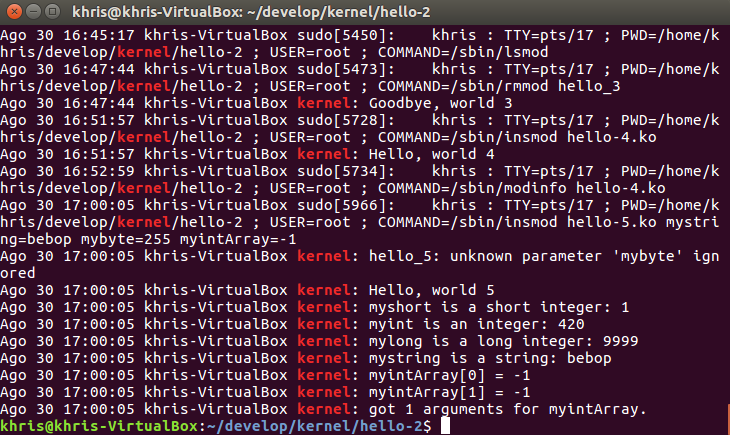
Resultado da execução do arquivo *“hello3-c”:*



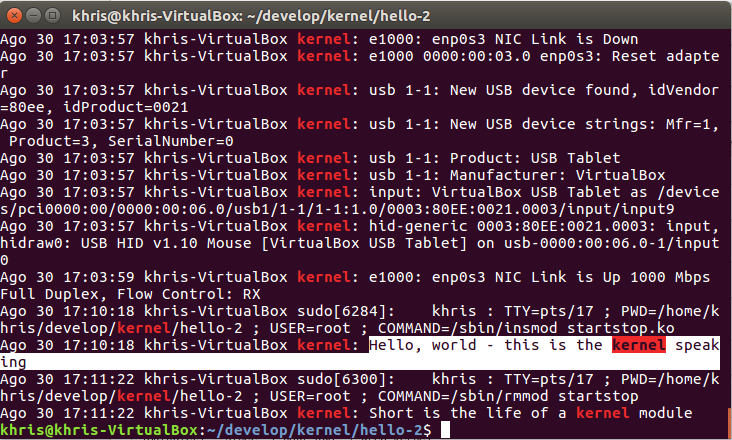
Resultado da execução do arquivo *“hello4-c”:*



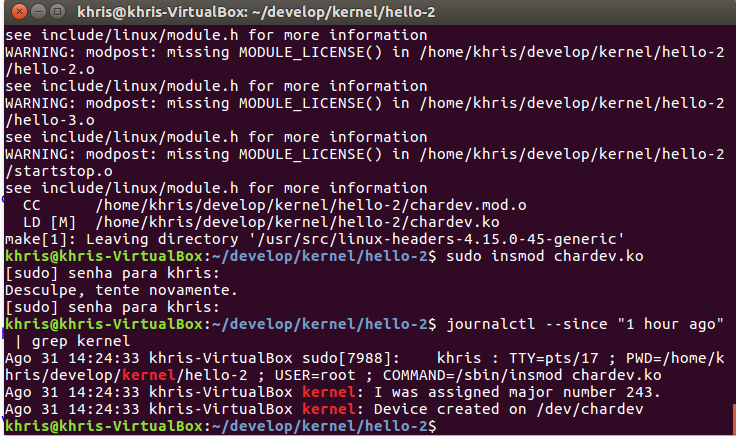
Resultado da execução do arquivo *“hello5-c”:*



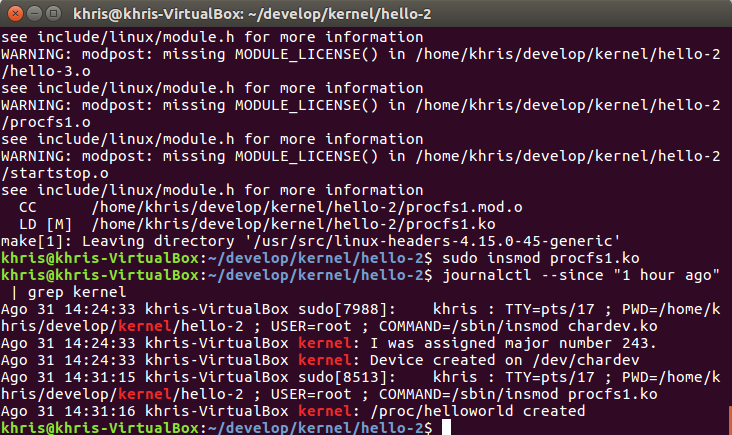
Resultado da execução do arquivo *“startstop.c”:*



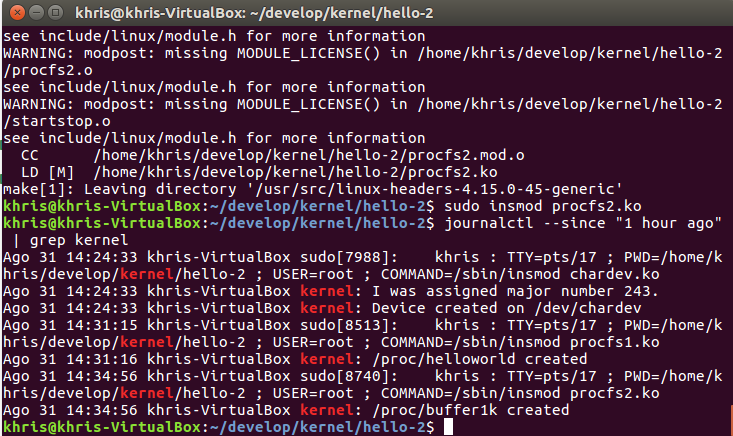
Resultado da execução do arquivo *“chardev.c”:*



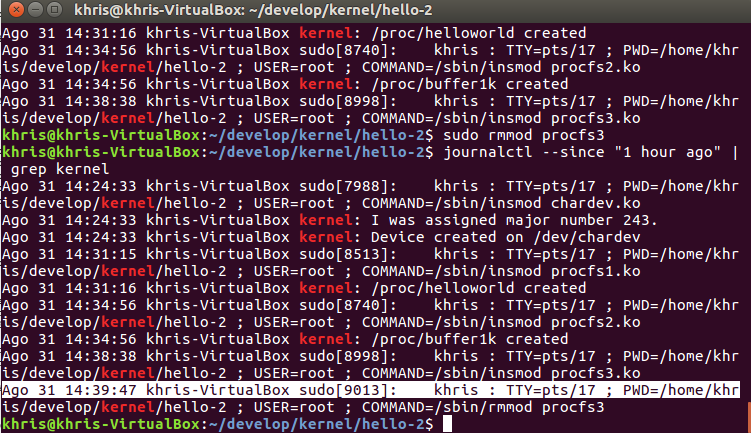
Resultado da execução do arquivo *“procfs1.c”:*



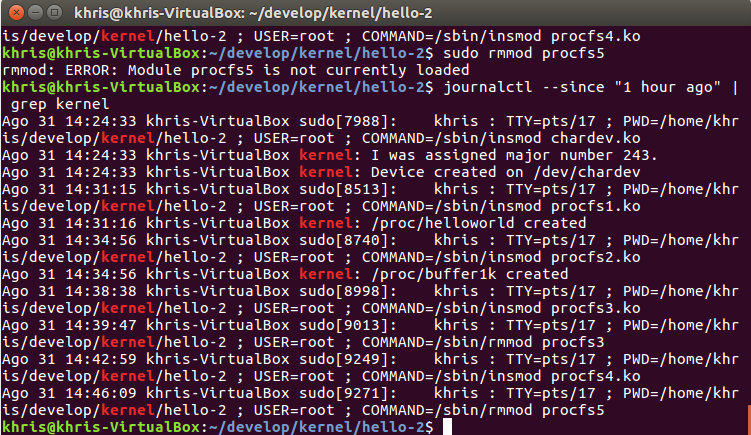
Resultado da execução do arquivo *“procfs2”:*



Resultado da execução do arquivo *“procfs3.c”:*



Resultado da execução do arquivo *“procfs4.c”:*



# **3. CONCLUSÃO**

Com esse experimento pudemos conhecer mais sobre o *linux* no que se diz respeito aos módulos de kernel, como criar, compilar, instalar e remover um módulo, habilidades que serão requisitadas no decorrer dos próximos experimentos. O objetivo principal do experimento de verificar, quase que minuciosamente, todos os processos referentes aos módulos de kernel antes apresentados, fora cumprido. Assim de forma geral inferimos que fora alcançado todos os objetivos propostos para esse experimento, tais como, entender de uma forma geral, os códigos dos módulos testes, entendendo o que eles propunham fazer e como era realizado, e aplicar a criação, compilação, instalação e remoção dos módulos testes.