

TUTORIAL:

CRIANDO UMA MÁQUINA VIRTUAL E COMPILANDO UM KERNEL

INTRODUÇÃO

Neste tutorial, irei ensinar a baixar e instalar o VMware Player, criar uma máquina virtual, instalar o sistema operacional (S.O.) (utilizarei o "CentOS 7") e compilar um kernel por meio de comandos no terminal. Feito por Nicollas Oliveira.

BAIXANDO E INSTALANDO O VMWARE PLAYER



Baixe o **VMware Player** no site:

"<https://www.vmware.com/br/products/workstation-player/workstation-player-evaluation.html>".

Vá em "**Fazer download agora**" para o sistema operacional que deseja instalar o **VMware** (neste tutorial, estou utilizando o S.O. **Windows**, por tanto, irei na opção do mesmo).

Aguarde o download ser concluído.

Execute o instalador.

Dê "**Next**".

Aceite os termos de uso (leia se caso preferir).

Dê "**Next**".

"**Next**" novamente.

Deixe as 2 caixas selecionadas por padrão:

- 1ª: **Checar por atualizações.**
- 2ª: **Juntar-se ao programa de melhoria da experiência do cliente.**

Dê "**Next**".

Deixe as 2 caixas selecionadas por padrão se preferir:

- 1ª: **Criar um atalho na área de trabalho do VMware Player.**
- 2ª: **Adicionar na pasta de programas do menu iniciar.**

Dê "**Next**".

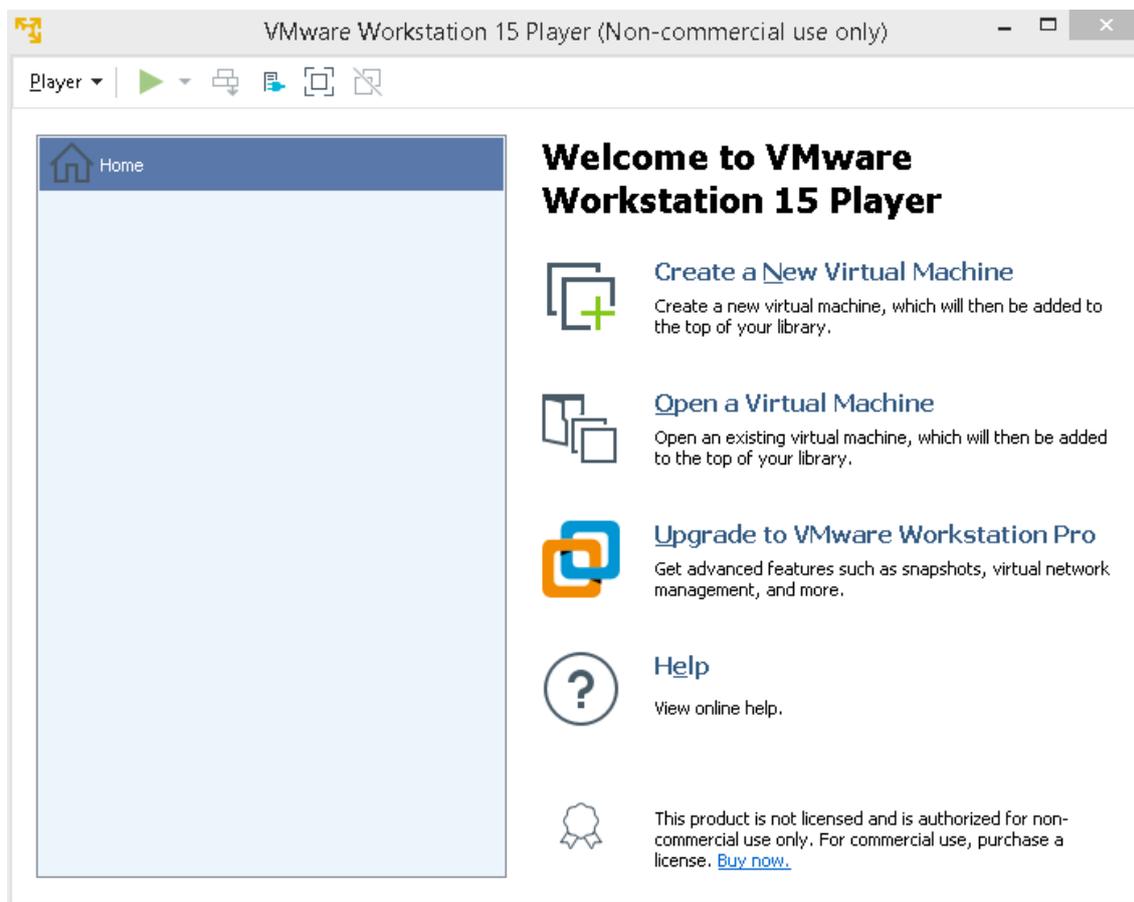
Clique em "**Install**".

Espera a instalação.

Por fim, clique em "**Finish**". Instalação concluída.



CRIANDO UMA MÁQUINA VIRTUAL



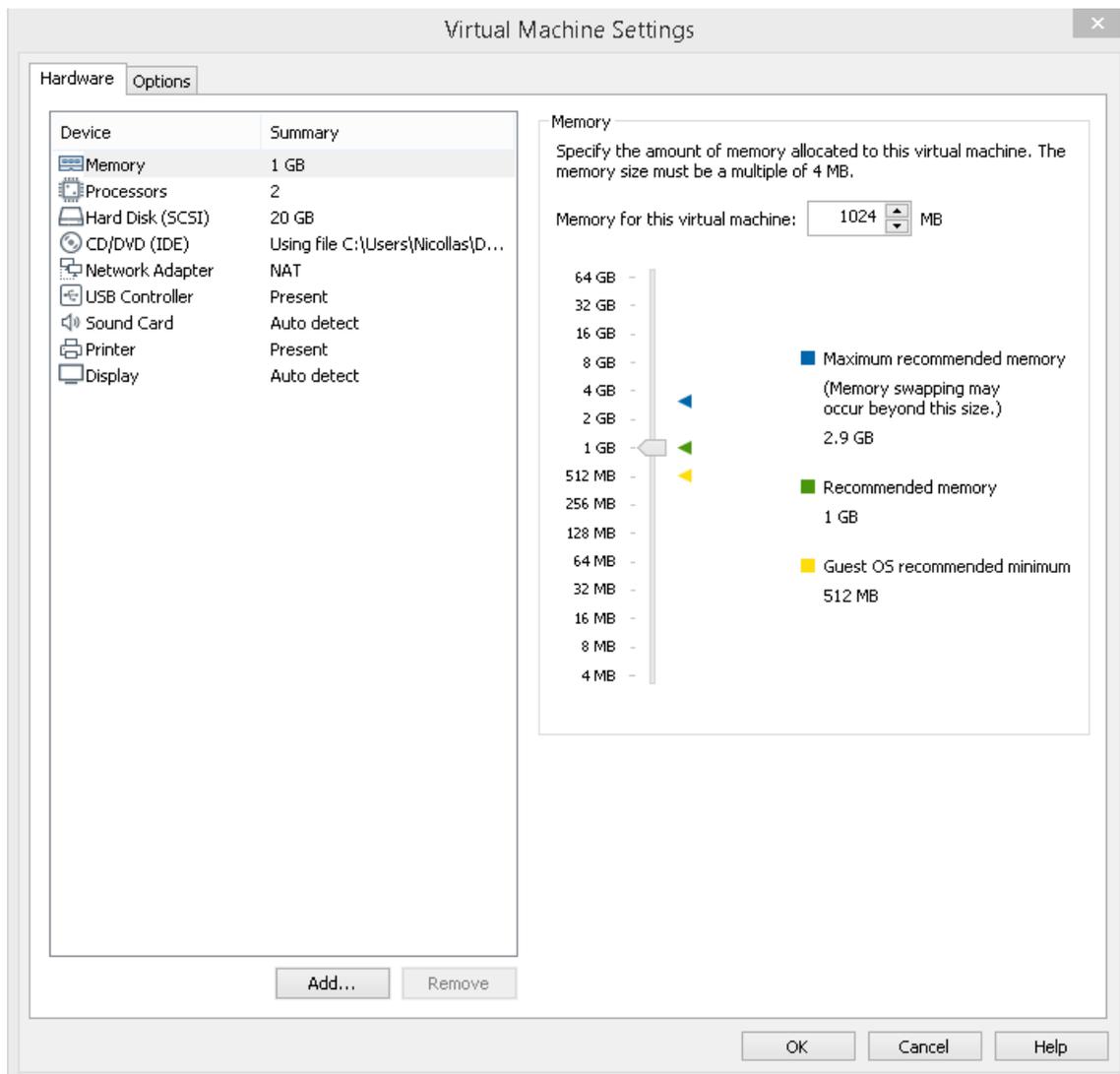
- Abra o **VMware Player** ("VMware Workstation 15 Player" no meu caso).
- Caso tenha uma licença, selecione a segunda opção, e insira a chave, se não, deixe a primeira opção selecionada e dê "Continue".
- Clique em "Finish".
- Caso abra uma janela de versões do programa, feche (ou leia se desejar).
- Irá exibir a interface do **VMware** (como na imagem acima).
- Clique em "Create a New Virtual Machine" (ou apenas **Ctrl + N**).
- Selecione o arquivo de imagem do S.O. (".iso"). Caso não esteja selecionado, vá em "Browse..." e procure pelo arquivo (irei usar o arquivo "CentOS-7-x86_64-Minimal-1810.iso").
- Digite o nome da máquina virtual.
- Insira a localização da máquina virtual dentro da máquina real (caso não esteja selecionado). Ou vá em "Browse..." e escolha um local.
- Dê "Next".
- Selecione o tamanho do disco virtual (em GB) (**recomenda-se 20GB**)

Selecione uma das opções:

- 1ª: Disco virtual armazenado como um único arquivo
- 2ª: Dividir disco virtual em múltiplos arquivos (mais fácil para mover a máquina virtual para outro computador, mas reduz a performance em discos muito grande) (recomendável).

Dê "Next".

Caso deseje configurar memória, adaptador de rede, quantidade de núcleos da CPU, etc. selecione "Customize Hardware...".

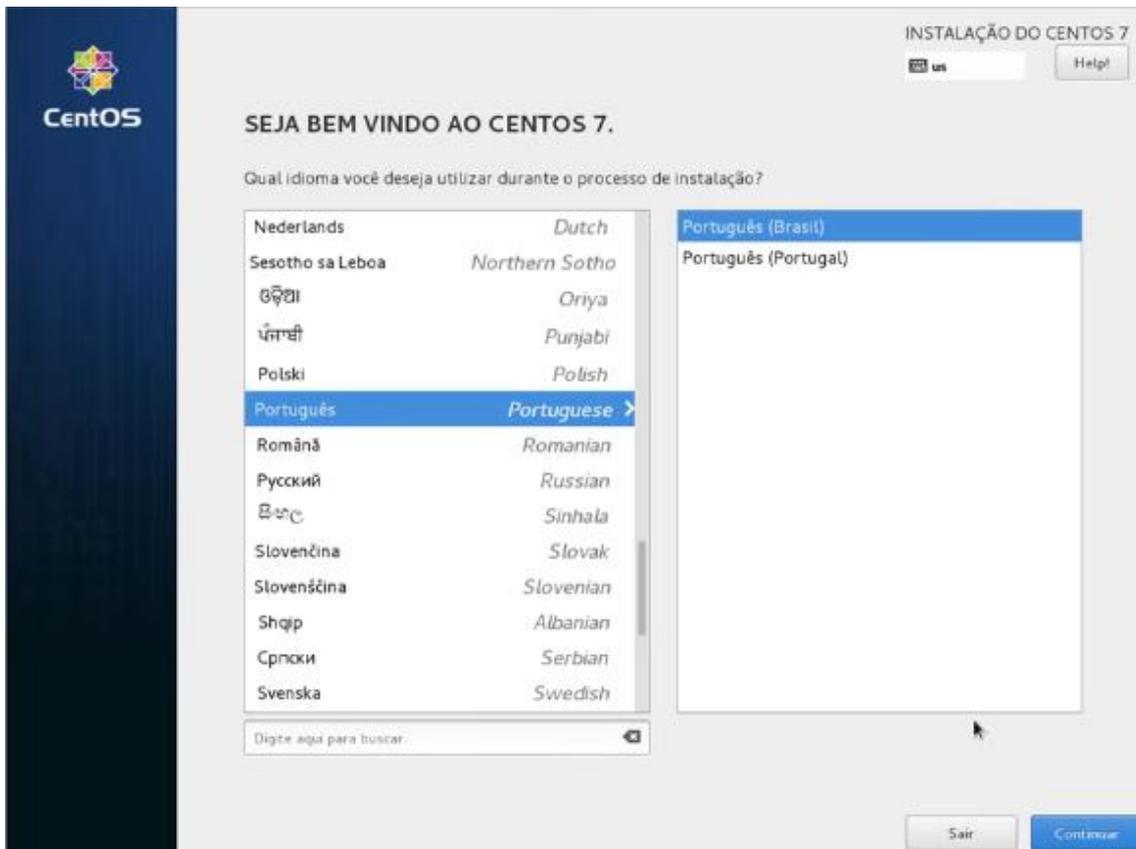


Deixe a caixa selecionada caso deseje iniciar a máquina virtual imediatamente. Clique em "Finish". Máquina virtual criada.

INSTALANDO O SISTEMA OPERACIONAL



Irá inicializar o S.O. (estou utilizando o "CentOS 7" neste tutorial).
Clique dentro da janela do S.O. para capturar o teclado.
Selecione (setas) "Install CentOS 7" e pressione ENTER.
Aguarde a inicialização da ISO do sistema.

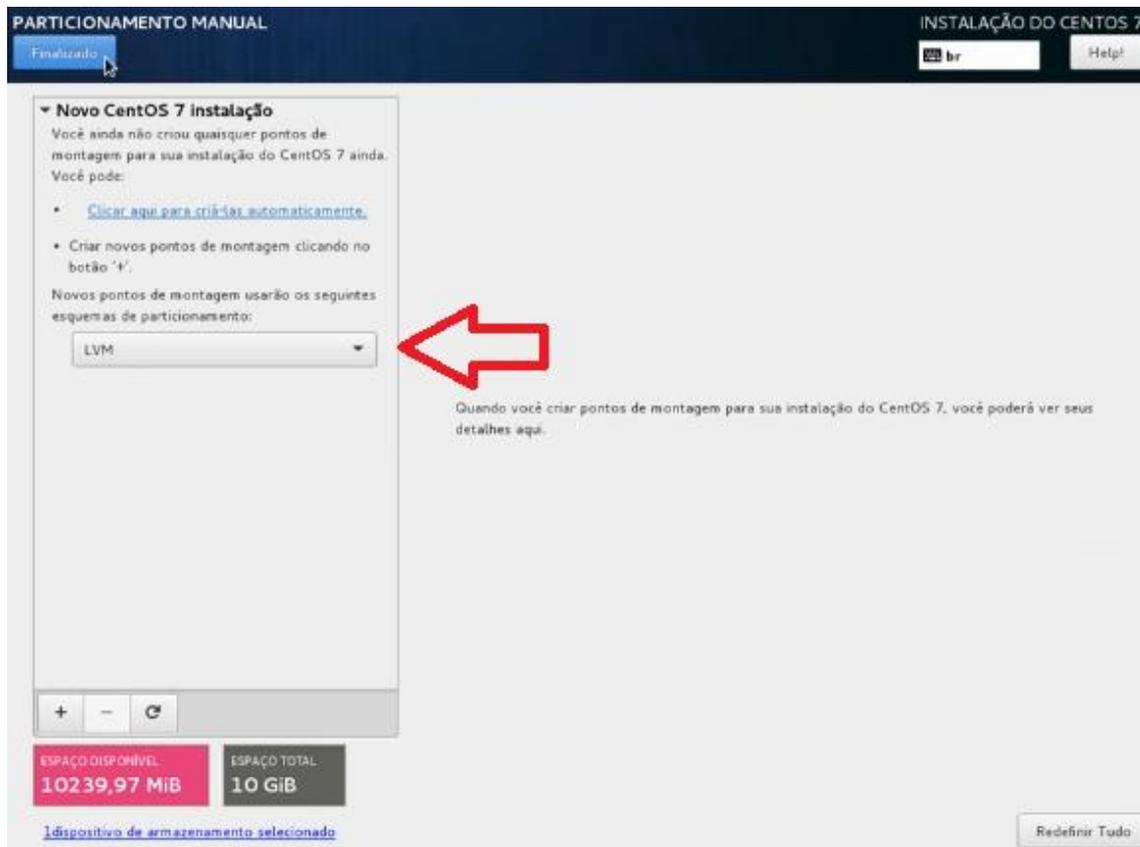


Selecione a linguagem que deseja utilizar durante o processo de instalação.
Clique em "Continuar".
Aguarde a identificação.
Se na "SELEÇÃO DE SOFTWARE", estiver "Instalações Mínimas", é porque foi utilizado a ISO do "CentOS Minimal".
Clique em "DESTINOINSTALAÇÃO".

Em "Outras opções de armazenamento" e "Particionamento", tem-se as opções de configurar automaticamente o particionamento, ou "Eu irei configurar o particionamento".

Neste tutorial, irei selecionar a segunda opção para configurar o particionamento (caso queira automaticamente, deixe a primeira opção marcada).

Clique em "Finalizado".



No lugar de "LVM" (clique em cima), selecione "Partição Padrão" e clique no "+" que está em baixo:

PontoMontagem: /

Capacidade Desejada: Tamanho desejado para o S.O. (coloquei 18GB como exemplo).

Clique em "Adicionar Ponto de Montagem"

Clique no "+" novamente e selecione no "PontoMontagem:" o "swap". Escolha o tamanho (vou deixar 300 Mb para o swap).

Clique em "Finalizado", em seguida em "Aceitar alterações".

Então, clique em "Iniciar Instalação".

Enquanto ele configura o sistema, pode-se configurar a senha do root e/ou criar um usuário.



Aguarde a finalização.

Clique em "Reinicializar".

Após isso, o S.O. é inicializado e está pronto para uso.

ENTRANDO NO SISTEMA OPERACIONAL E BAIXANDO O KERNEL

```
CentOS Linux 7 (Core)
Kernel 3.10.0-957.el7.x86_64 on an x86_64

localhost login: _
```

Logue como `root`, e digite o comando `#dhclient` para configurar a rede.

Após isso, digite `#yum update` para atualizar os pacotes. Caso pergunte, digite "y", dê `ENTER` e aguarde.

Digite o comando `#yum install -y bc vim wget gcc bison ncurses-devel make gcc-c++ git openssl-devel` para instalar os pacotes necessários para compilar o kernel.

Em seguida, digite `#wget https://www.kernel.org/pub/linux/kernel/v3.x/linux-3.19.tar.gz` para baixar o kernel utilizado neste tutorial (linux-3.19).

Digite o comando `#ls` para listar os arquivos encontrados, e mova o arquivo baixado (com a extensão `".tar.gz"`) para a pasta `"/usr/src/kernels"` com o comando `#mv NOME_DO_ARQUIVO /usr/src/kernels/` substituindo o `"NOME_DO_ARQUIVO"` pelo nome do arquivo listado anteriormente.

Digite comando `#cd /usr/src/kernels/` para entrar na pasta do arquivo.

Digite `#tar -xvzf NOME_DO_ARQUIVO` para descompactar o código fonte do kernel (lembrando que no lugar de `"NOME_DO_ARQUIVO"`, será inserido o nome do arquivo que foi listado (`".tar.gz"`)).

```
[root@localhost ~]# ls
anaconda-ks.cfg  linux-3.19.tar.gz
[root@localhost ~]# mv linux-3.19.tar.gz /usr/src/kernels
[root@localhost ~]# cd /usr/src/kernels
[root@localhost kernels]# ls
linux-3.19.tar.gz
[root@localhost kernels]# tar -xvzf linux-3.19.tar.gz _
```

Digite `#ls` novamente para listar os arquivos do diretório.

Entre na pasta que foi descompactada com `#cd NOME_DA_PASTA` onde no lugar de `"NOME_DA_PASTA"` será inserido o nome do arquivo sem o `".tar.gz"`

Digite `#date +"-%Y%m%d%H"` para gerar uma string de extraversión (que será exibida logo abaixo, grave esta string).

Digite o comando `#vim Makefile` para editar o arquivo `"Makefile"`.

```
[root@localhost kernels]# ls
linux-3.19  linux-3.19.tar.gz
[root@localhost kernels]# cd linux-3.19
[root@localhost linux-3.19]# ls
arch          drivers      Kbuild      mm           REPORTING-BUGS  sound        vmlinux.o
block        firmware    Kconfig     modules.builtin  samples         System.map  x509.genkey
COPYING      fs          kernel      modules.order  scripts         tools
CREDITS     include     lib         Module.symvers  security        usr
crypto      init        MAINTAINERS net           signing_key.priv  virt
Documentation ipc         Makefile    README        signing_key.x509  vmlinux
[root@localhost linux-3.19]# vim Makefile
```

Na frente de `"EXTRAVERSION = "`, digite (pressione a tecla `"i"` para editar) o arquivo, e insira a string gerada pelo comando `#date` dado anteriormente (ex: `"-20190520c1"`).

```
VERSION = 3
PATCHLEVEL = 19
SUBLEVEL = 0
EXTRAVERSION = -20190520c1
NAME = Diseased Newt

# *DOCUMENTATION*
# To see a list of typical targets execute "make help"
# More info can be located in ./README
# Comments in this file are targeted only to the developer, do not
# expect to learn how to build the kernel reading this file.

# Do not use make's built-in rules and variables
# (this increases performance and avoids hard-to-debug behaviour);
MAKEFLAGS += -rR

# Avoid funny character set dependencies
unexport LC_ALL
LC_COLLATE=C
LC_NUMERIC=C
export LC_COLLATE LC_NUMERIC

# Avoid interference with shell env settings
unexport GREP_OPTIONS

# We are using a recursive build, so we need to do a little thinking
# to get the ordering right.
#
# Most importantly: sub-Makefiles should only ever modify files in
# their own directory. If in some directory we have a dependency on
# a file in another dir (which doesn't happen often, but it's often
# unavoidable when linking the built-in.o targets which finally
# turn into vmlinux), we will call a sub make in that other dir, and
# after that we are sure that everything which is in that other dir
# is now up to date.
#
"Makefile" 1592L, 54338C
```



Feito isso, pressione **ESC**, digite **":wq"** e dê **ENTER**.
Digite **#make mrproper** para dar uma limpeza no diretório da fonte do kernel.

CONFIGURANDO O KERNEL

Para editar a configuração do kernel, digite o comando `#make menuconfig`. Agora, será alterado algumas configurações do kernel (movimente-se pelo menu (setas)).

```
.config - Linux/x86 3.19.0-20190520c1 Kernel Configuration
Linux/x86 3.19.0-20190520c1 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenu ----).
Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes
features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in
[ ] excluded <M> module < > module capable

[*] 64-bit kernel
  General setup --->
  [*] Enable loadable module support --->
  -* Enable the block layer --->
  Processor type and features --->
  Power management and ACPI options --->
  Bus options (PCI etc.) --->
  Executable file formats / Emulations --->
  [*] Networking support --->
    Device Drivers --->
    Firmware Drivers --->
    File systems --->
    Kernel hacking --->
    Security options --->
  -* Cryptographic API --->
  [*] Virtualization --->
    Library routines --->

<Select> < Exit > < Help > < Save > < Load >
```

*Legenda: **Item: Local do item desejado / sublocal <tecla pressionada>**

1: **Networking support / Bluetooth subsystem support <N>**

Desabilitado o suporte de bluetooth, pois não há um dispositivo físico para o mesmo.

2: **Networking support / Wireless / cfg80211 <N>**

Desabilitado o suporte de rede sem fio, pois não há um dispositivo físico para o mesmo.

3: **Networking support / NFC subsystem support <M>**

Ativado o módulo de suporte ao NFC para a comunicação entre dispositivos de curto alcance.

4: **Networking support / IrDA (infrared) <Y>**

Ativado o suporte ao infravermelho para a comunicação entre dispositivos utilizando o feixe de luz infravermelho.

5: **Networking support / Amateur Radio support <Y>**

Ativado o suporte ao rádio amador.

6: **File system / DOS/FAT/NT Filesystems / NTFS file system support <Y>**

Ativado o suporte ao tipo de arquivo NTFS (sistema de arquivos padrão do Windows NT).

7: Device Drivers / Sound card support <N>

Desativado o suporte à placa de som, pois o mesmo não será usado.

8: Device Drivers / Accessibility support <Y>

Ativado o suporte à acessibilidade.

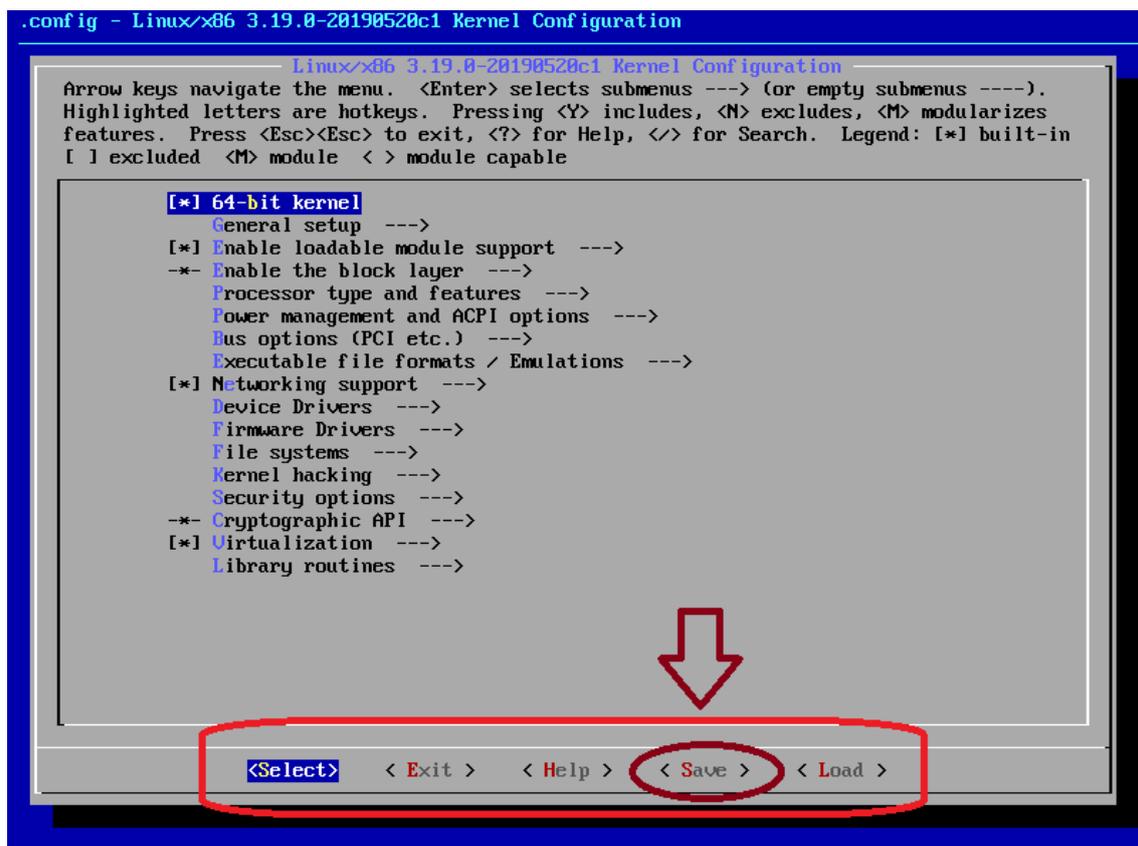
9: Device Drivers / Real Time Clock <N>

Desativado o relógio em tempo real.

10: File system / CD-ROM/DVD Filesystems / ISO 9660 CDROM file system support <N>

Desativado o suporte ao tipo de arquivo ISO do CDROM (disco óptico).

Após isso, vá em "Save" (setas (direita e esquerda)), dê ENTER, e ENTER novamente (OK).



Ao voltar para o terminal, digite o comando `#make` (caso haja um processador Dual Core na máquina que está compilando o kernel, use o comando `#make -j4` (A opção "-j" executa N tarefas por segundo, onde "N" será a quantidade de funcionalidades compiladas por segundo, tornando mais rápido))

Aguarde a compilação do kernel.

Após compilado, digite o comando `#make modules_install` para instalar os módulos do kernel e gerar a imagem `initramfs (initrd)`. Ele irá copiar todos os módulos externos para o diretório `"/lib/modules/"`.

Por fim, execute o comando `#make install` para a instalação.

Kernel compilado com sucesso, agora, basta testar e usufruir do mesmo.

CentOS Linux (3.19.0-20190520c1) 7 (Core)

CentOS Linux (3.10.0-957.el7.x86_64) 7 (Core)

CentOS Linux (0-rescue-7f0bc0322ca443048b13eaa6c51f503a) 7 (Core)

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.