

Sistemas Operacionais

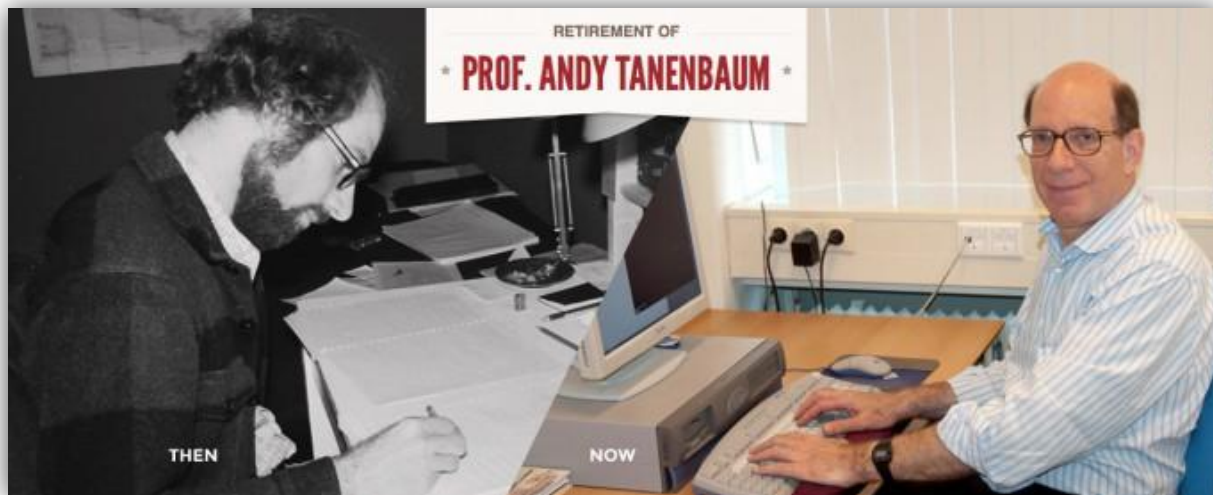
Linux: Conceitos Básicos:

- Kernel Linux
- CoreUtils



Linux - Origem

- Originalmente, o Prof. **Andrew S. Tanenba** criou em Amsterdam (Vrije Universiteit) um S.O. chamado Minix para explicar os princípios de S.O. para seus alunos. Tinha 12.000 linhas de código de códigos assembly e C.
- Minix foi distribuído com o código fonte MINIX em disquetes e com um manual de referência. Sua estrutura era compatível com UNIX.



Linux - Origem

- Linus Torvalds, começou a estudar ciência da computação na Universidade de Helsinki, na Finlândia. Decidiu criar a sua própria implementação de um terminal em seu recém obtido computador 80386, principalmente para acessar o servidor Unix da instituição de ensino.
- A intenção de Torvalds era a de fazer o projeto rodar especificamente em sua máquina 80386, com o desenvolvimento sendo feito a partir do Minix.
- Em 1991, Linus Torvalds decidiu divulgar abertamente o projeto através da Usenet (uma espécie de antecessora da internet baseada em troca de mensagens) pedindo sugestões e colaborações para a sua iniciativa.



Linux - Origem

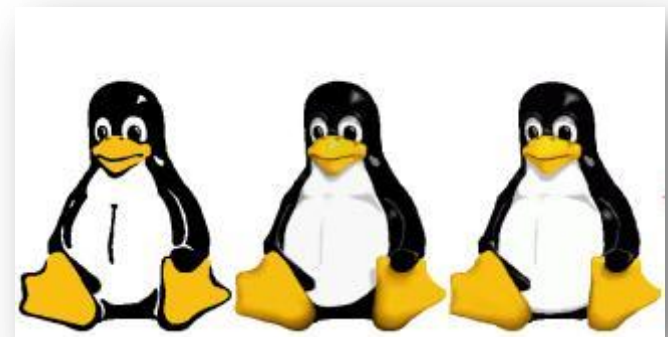
- Inicialmente não tinha um nome. Torvalds atribuiu ao kernel a denominação **Freax**, uma mistura de *free* (livre) com *freak* (monstruoso, esquisito) e a letra 'x', para lembrar o Unix.
- Um programador **Ari Lemmke**, decidiu criar um servidor de FTP que hospedaria o software uma pasta de nome "**linux**" (muito provavelmente, uma mistura de Linus com Unix). Essa nomenclatura foi aceita pela comunidade.
- O Linux é um kernel. Os demais softwares necessários para que tenhamos um SO efetivamente vem do projeto GNU.

Linux - Origem

- GNU é a sigla para um nome curioso: "***GNU is Not Unix (GNU Não é Unix)***". Era um projeto que tentava criar um SO Unix, criado em 1984.
- Com o passar dos anos, o projeto foi ganhando recursos, como compiladores e editores de texto. Mas, faltava um elemento importantíssimo: **Um kernel.**
- O Linux que temos hoje é formado por um kernel (Linux) e um conjunto com software GNU.

Linux - Origem

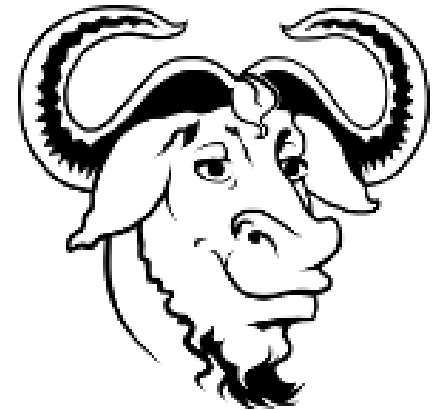
- Linus Torvalds em um debate para escolha do logotipo afirmou em uma mensagem que gostava muito de Pinguins, sem motivo nenhum, aparentemente.
- O uso de uma figura de um Pinguin como logotipo do Linux certamente representa algo algo feito em comunidade, pois "pinguins vivem em bandos".
- O Mascote foi nomeado de “Tux”, referência ao nome de Linus Torvalds com Unix: **T**orvalds **U**ni**X**.



Linux - Origem

Linux é Open Source.

Isso significa que **os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software.**



Assim sendo, “software livre” é uma questão de liberdade, não de preço.

Ver licenças: <https://www.gnu.org/licenses/licenses.html>

Linux - Origem

Uma distribuição Linux (geralmente abreviada de distro) é um SO criado a partir de uma coleção de softwares, com o uso do núcleo Linux, um gerenciador de pacotes e um repositório de programas.

Na maior parte dos casos, as distribuições Linux utilizam bibliotecas e utilidades criadas pela GNU porém há distribuições Linux para uma variedade de casos de uso, desde sistemas embarcados, computadores pessoais, para supercomputadores, etc.

Há mais de 600 distribuições Linux existentes (a maioria está em desenvolvimento ativo).

Existem distribuições com o suporte de companhias pagas e distribuições completamente suportadas por suas comunidades.

Linux – Ambiente de Trabalho

- O **shell** é um interpretador de comandos. Ele interage com o sistema operacional, fazendo a ligação entre os comandos que você digita e as atividades que o kernel pode realizar.
- O propósito do **shell** é tornar o sistema operacional mais amigável, por ser muito mais fácil para o usuário lembrar o nome do comando que o conjunto de chamadas de sistema que estão por trás dele.
- O Linux possui mais do que um **shell**, cada qual com características ligeiramente diferentes e funções especiais. Você pode usar o **shell** que preferir.
- O **shell** padrão para o Linux é o Bourne Again Shell (bash). Entre outros, tem o C-shell (csh), Korn shell (ksh), etc.

Linux – Ambiente de Trabalho

- O superusuário **root** é o nome da conta que tem acesso completo a todos os componentes do sistema.
- Normalmente, a conta de superusuário é somente utilizada na execução de tarefas de administração do sistema, como a criação de novas contas, desligar o sistema, etc.

`[root@aulas] #`

ou normal:

`[estudante@aulas] $`

***COMANDO SU** – este comando torna a sessão atual em superusuário ou outro usuário.*

Linux – Estrutura de Diretórios

<https://debian-handbook.info/browse/pt-BR/stable/sect.filesystem-hierarchy.html>

*Um sistema organizado de acordo com o **Filesystem Hierarchy Standard (FHS)**,
define a finalidade de cada diretório.*

- /bin/: programas básicos;
- /boot/: núcleo Linux e outros arquivos necessários para os primeiros passos de seu processo de inicialização;
- /dev/: arquivos de dispositivo;
- /etc/: arquivos de configuração;
- /home/: arquivos pessoais dos usuários;
- /lib/: bibliotecas básicas;
- /media/*: pontos de montagem para dispositivos
- /mnt/: ponto de montagem temporário;
- /opt/: aplicações extras fornecidas por terceiros;
- /root/: arquivos pessoais do administrador (root);
- /run/: dados de execução volátil (volatile runtime data) que não persistem entre re-inicializações;

Linux – Estrutura de Diretórios

(<https://debian-handbook.info/browse/pt-BR/stable/sect.filesystem-hierarchy.html>)

- /sbin/: programas do sistema;
- /srv/: dados utilizados por servidores hospedados neste sistema;
- /tmp/: arquivos temporários, este diretório é comumente limpo na inicialização;
- /usr/: aplicações; este diretório é subdividido em bin, sbin, lib (de acordo com a mesma lógica do diretório raiz). Além disso, /usr/share/ contém dados independentes de arquitetura.
/usr/local/ é feito para ser usado pelo administrador para instalar aplicativos manualmente, sem sobrescrever arquivos administrados pelo sistema de empacotamento (dpkg).
- /var/: dados variáveis manipulados por daemons. Isto inclui arquivos de log, filas, spools, caches e por aí vai.
- /proc/ e /sys/ são específicos do núcleo Linux (e não fazem parte do FHS). Eles são usados pelo núcleo para exportar dados para o espaço de usuário

LINUX – COMANDOS BÁSICOS

Linux – Comandos Básicos

Durante o login a podem ser executados os seguintes arquivos de configuração :

~/ .bashrc
/etc/bashrc

Quando o shell é invocado em modo login shell ele lê os quatro arquivos de profile. Ou então quando se executa uma nova sessão são lidos os arquivos:

~/ .bashrc
/etc/bashrc

Da mesma forma que existe os arquivos de login, existe também o arquivo de logout:

~/ .bash_logout

O histórico é armazenado no arquivo de usuário **bash_history**

O Diretório **/etc/skel** é o modelo de configuração usado sempre que um usuário é criado.

Comandos Básicos: ls

Exibir Arquivos :

```
[estudante@aulas]$ ls
```

Tente:

```
ls -la
```

```
ls -A
```

```
ls -lh
```

```
ls -lt
```

```
ls -li
```

```
ls -lsh
```

```
ls -la | grep ^d
```



Linux – Comandos Básicos: cat

Mostra o conteúdo de um arquivo.

```
[estudante@aulas]$ cat <nomeArquivo>
```

Criar um arquivo:

```
cat /etc/passwd > senhas.txt
```

Opções:

> abre o arquivo como entrada

< abre o arquivo como saída

>> acrescenta dados em um arquivo já existente

- ***tac** é um alternativa para mostrar um arquivo de do final para o início.*

Linux – Comandos Básicos: CAT

Mostra o conteúdo de um arquivo.

```
[estudante@aulas]$ cat /etc/passwd
```

Criar um arquivo:

```
cat "arquivo" > "nome_do_arquivo"
```

Opções:

> abre o arquivo como entrada

< abre o arquivo como saída

>> acrescenta dados em um arquivo já existente



Linux – Comandos Básicos: tail e head

TAIL

Mostra o final de um arquivo;

- f Mantém o arquivo aberto com atualização instantânea;
- n Define o número de linhas;

HEAD

Mostra o início de um arquivo;

- c Especifica o número de bytes que será mostrado;
- n Define o número de linhas;

TENTE:

```
$tail /var/log/dmesg  
$head -n1 /var/log/dmesg  
$tail -f /var/log/dmesg
```



Linux – Comandos Básicos: last e cut

LAST

Mostra o log de login;

CUT

Remove dados de arquivo

```
cut -f<coluna> -d<delimitador> arquivo
```

Tente:

```
last > dados.txt  
cut -f1 -d" " dados.txt  
cut -c1-9 dados.txt
```

Linux – Comandos Básicos: vi

VI

Editor de texto.

Opções:

- :q! Sai sem salvar;
- :x Sai e salva;
- :w <nome Salva com o nome especificado;
- <insert> possibilita a inserção de dados no VI;
- :/ Localiza palavra do início para o fim (n para a próxima)
- :? Localiza palavra do fim para o início (n para a próxima)
- :% s/poa/POA Substitui em todo o arquivo a ocorrência poa por POA.

Linux – Comandos Básicos: grep

grep – Localiza expressões em arquivos.

`grep [opção] “expressão”`

Opções:

- i: ignora a diferença entre maiúscula e minúscula
- c: mostra no. de vezes que a expressão foi encontrada
- l: lista somente o nome dos arquivos que contém a expressão procurada
- n: numera cada linha que contém a expressão procurada

Exemplo:

- `cd /etc`
- `grep -i root *`

Tente:

`grep -n -i ifrs *`

Linux – Comandos Básicos: grep

grep – Localiza expressões em arquivos.

grep [opção] “expressão”

Opções:

- i: ignora a diferença entre maiúscula e minúscula
- c: mostra no. de vezes que a expressão foi encontrada
- l: lista somente o nome dos arquivos que contém a expressão procurada
- n: numera cada linha que contém a expressão procurada

Tente:

```
grep -n -i ifrs *  
who | grep aluno  
ls | grep net  
cat /etc/passwd | grep admin
```

Linux – Comandos Básicos: wc

wc – conta linhas, palavras e caracteres

wc arquivo

Opções:

Tente:

```
who | grep aluno | wc -l
```

```
ls | grep net | wc -l
```

```
cat /etc/passwd | grep admin | wc -l
```

Linux – Comandos Básicos: wget

wget – downloader não interativo.

wget <opções> URL

Tente:

```
wget https://ww4.al.rs.gov.br
```

```
wget https://ww4.al.rs.gov.br -O saida.html
```

```
wget -v https://ww4.al.rs.gov.br -O saida.html
```

```
wget https://portal.al.go.leg.br/legado/deputado/ver/id/1222
```




Linux – Comandos Básicos: chmod

chmod

altera a permissão de acesso a arquivos e diretórios.

U	User;
G	Group;
O	Other;
A	All;
+	Ativa um direito;
-	Desativa um direito;
=	Igual a um direito;

Exemplo:

```
chmod u+x arquivo
```

```
chmod u-w go=r *.zip -r
```

Kernel Linux (Compilação)

Por que Compilar Kernel?

Normalmente não é preciso compilar um Kernel Linux, considerando que ele irá funcionar de maneira adequada na grande maioria de hardwares disponível.

Existem várias versões de Kernel, entre eles o LTS (Long -Term Support)

Os kernels de suporte de longo prazo ("LTS") são marcados como "estáveis".

Por que Compilar Kernel?

No entanto, poderá ser útil compilar um novo kernel para:

- **Tratar de necessidades** específicas de hardware (conflitos com os kernels pré-compilados)
- **Ativar/Desativar opções** do kernel que não são suportadas pelos kernels pré-fornecidos (suporte para muita memória, sistema de arquivos, drivers, etc)
- **Otimizar o kernel** removendo drivers inúteis para diminuir o tempo de inicialização e/ou tamanho
- **Criar um kernel monolítico** em vez de um kernel modular
- Usar um **kernel de desenvolvimento**
- **Aprender mais** sobre Sistemas Operacionais

Compilação Kernel Linux

Download do
Código Fonte

- (www.kernel.org)

Configuração dos
Módulos do Kernel

- (`#make menuconfig`)

Geração da Imagem

- (`#make bzImage`)

Compilação do
Módulos

- (`#make modules`)

Instalação dos
Módulos

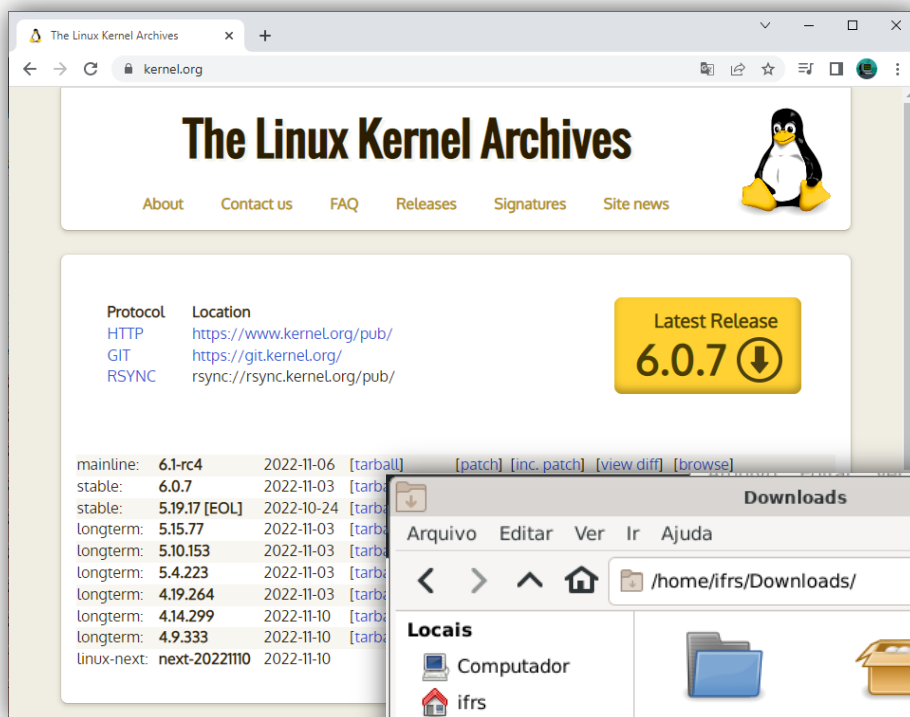
- (`#make install`)

Configuração do
GRUB (inicialização)

- (`#update-grub`)



Download do Código Fonte




The Linux Kernel Archives

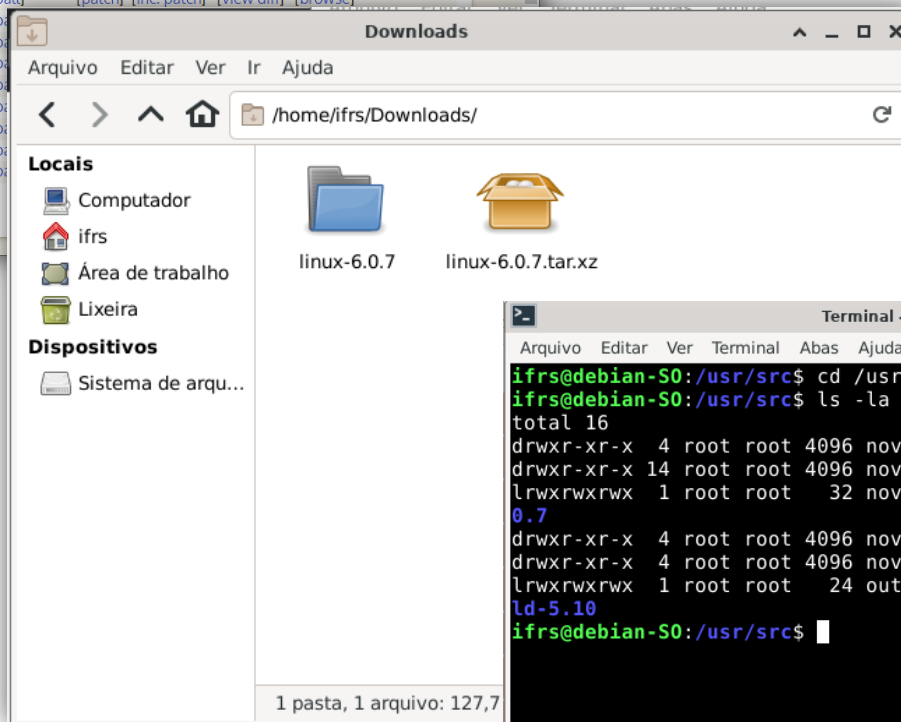
About Contact us FAQ Releases Signatures Site news

Protocol Location

HTTP	https://www.kernel.org/pub/
Git	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/



Latest Release
6.0.7 

mainline:	6.1-rc4	2022-11-06	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]
stable:	6.0.7	2022-11-03	[tarball]				
stable:	5.19.17 [EOL]	2022-10-24	[tarball]				
longterm:	5.15.77	2022-11-03	[tarball]				
longterm:	5.10.153	2022-11-03	[tarball]				
longterm:	5.4.223	2022-11-03	[tarball]				
longterm:	4.19.264	2022-11-03	[tarball]				
longterm:	4.14.299	2022-11-10	[tarball]				
longterm:	4.9.333	2022-11-10	[tarball]				
linux-next:	next-20221110	2022-11-10	[tarball]				



Downloads

Arquivo Editar Ver Ir Ajuda

< > ^  /home/ifrs/Downloads/ 

Locais

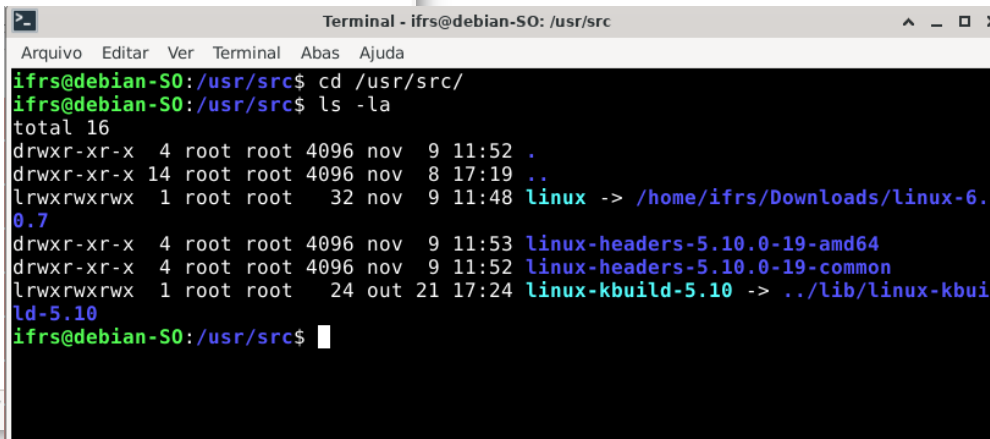
- Computador
- ifrs
- Área de trabalho
- Lixeira

Dispositivos

- Sistema de arqu...

linux-6.0.7 linux-6.0.7.tar.xz

1 pasta, 1 arquivo: 127,7



```
Terminal - ifrs@debian-SO: /usr/src
Arquivo Editar Ver Terminal Abas Ajuda
ifrs@debian-SO:/usr/src$ cd /usr/src/
ifrs@debian-SO:/usr/src$ ls -la
total 16
drwxr-xr-x  4 root root 4096 nov  9 11:52 .
drwxr-xr-x 14 root root 4096 nov  8 17:19 ..
lrwxrwxrwx  1 root root   32 nov  9 11:48 linux -> /home/ifrs/Downloads/linux-6.0.7
drwxr-xr-x  4 root root 4096 nov  9 11:53 linux-headers-5.10.0-19-amd64
drwxr-xr-x  4 root root 4096 nov  9 11:52 linux-headers-5.10.0-19-common
lrwxrwxrwx  1 root root   24 out 21 17:24 linux-kbuild-5.10 -> ../lib/linux-kbuild-5.10
ifrs@debian-SO:/usr/src$
```



Configurações dos Módulos do Kernel

```
# cd usr/src/<diretorioCodigoFonte>
# make menuconfig
```

```
Terminal - ifrs@debian-S0: /usr/src/linux
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@debian-S0:/# cd /usr/src/linux
root@debian-S0:/usr/src/linux# make menuconfig
```

```
.config - Linux/x86 6.0.7ifrs Kernel Configuration

Linux/x86 6.0.7ifrs Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). 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

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

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

Configurações dos Módulos do Kernel

```
.config - Linux/x86 6.0.7ifrs Kernel Configuration

Linux/x86 6.0.7ifrs Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus --
letters are hotkeys. Pressing <Y> includes, <N> excludes,
<Esc><Esc> to exit, <?> for Help, </> for Search. Legend:
< > module capable

General setup --->
[ ] 64-bit kernel
Processor type and features --->
[*] Mitigations for speculative execution vulnerabilities
Power management and ACPI options --->
Bus options (PCI etc.) --->
Binary Emulations ----
[*] Virtualization --->
```

Editar o arquivo Makefile

Alterar EXTRAVERSION

#nano Makefile

Ou

#vi Makefile

```
Arquivo Editar Ver Terminal Abas Ajuda
GNU nano 5.4 Makefile
# SPDX-License-Identifier: GPL-2.0
VERSION = 6
PATCHLEVEL = 0
SUBLEVEL = 7
EXTRAVERSION =ifrs
NAME = Ubuntu 6.0.7ifrs Linux

# *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.

$(if $(filter __%, ${MAKECMDGOALS}), \
    $(error targets prefixed with '__' are only for internal use))

# That's our default target when none is given on the command line
```


Geração da Imagem

```
# make bzImage
```

Cria uma imagem compactada do kernel do kernel compilado.

Obs. bzImage não é compactado com bzip.

O arquivo criado (imagem) **bzImage** fica armazenado em **/usr/src/linux/arch** or **arch/i386/boot**

Obs.: arch = arquitetura (32 ou 64bits)

Geração da Imagem

```
# make bzImage
```

```
Terminal - ifrs@debian-S0: /usr/src/linux
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
root@debian-S0:/usr/src/linux# ls -lah arch/x86/boot/
total 12M
drwxr-xr-x  4 ifrs ifrs 4,0K nov 10 13:47 .
drwxr-xr-x 28 ifrs ifrs 4,0K nov 10 13:09 ..
-rw-r--r--  1 ifrs ifrs 3,4K nov  3 12:00 a20.c
-rw-r--r--  1 root root 11K nov 10 13:47 a20.o
-rw-r--r--  1 root root 6,9K nov 10 13:47 .a20.o.cmd
-rw-r--r--  1 ifrs ifrs 1,9K nov  3 12:00 apm.c
-rw-r--r--  1 root root 1,5K nov  9 13:06 bioscall.o
-rw-r--r--  1 root root 1,2K nov  9 13:06 .bioscall.o.cmd
-rw-r--r--  1 ifrs ifrs 1,4K nov  3 12:00 bioscall.S
-rw-r--r--  1 ifrs ifrs 1,1K nov  3 12:00 bitops.h
-rw-r--r--  1 ifrs ifrs 7,5K nov  3 12:00 boot.h
-rw-r--r--  1 root root 5,4M nov 10 13:47 bzImage
-rw-r--r--  1 root root 152 nov 10 13:47 .bzImage.cmd
-rw-r--r--  1 ifrs ifrs 3,2K nov  3 12:00 cmdline.c
-rw-r--r--  1 root root 7,4K nov 10 13:47 cmdline.o
-rw-r--r--  1 root root 7,0K nov 10 13:47 .cmdline.o.cmd
drwxr-xr-x  2 ifrs ifrs 4,0K nov 10 13:47 compressed
-rw-r--r--  1 root root 1,5K nov 10 13:47 copy.o
-rw-r--r--  1 root root 2,7K nov 10 13:47 .copy.o.cmd
-rw-r--r--  1 ifrs ifrs 1,1K nov  3 12:00 copy.S
-rw-r--r--  1 ifrs ifrs 2,0K nov  3 12:00 cpu.c
-rw-r--r--  1 ifrs ifrs 6,0K nov  3 12:00 cpucheck.c
-rw-r--r--  1 root root 19K nov 10 13:47 cpucheck.o
-rw-r--r--  1 root root 7,2K nov 10 13:47 .cpucheck.o.cmd
-rw-r--r--  1 ifrs ifrs 2,9K nov  3 12:00 cpuflags.c
```



Compilação dos Módulos do Kernel

```
# make modules
```

Obs.:

- Os módulos são gravados em /usr/lib/modules/versão_do_kernel
- Operação bastante demorada.

Instalação dos Módulos

```
# make install
```

Obs.:

- O comando **make install** irá copiar o novo kernel compilado, bibliotecas e documentação para os diretórios de destinos.

Configuração do GRUB

```
# cd /boot  
  
# mkinitramfs -o (kernel.img) (kernel)  
  
# update-grub  
  
# reboot
```

Obs.

- Caso o kernel “Novo” não aparece no menu de inicialização do GRUB, selecione o modo avançado para listar todas os *Kernels*.

GNU Core Utilities.

GNU - Origem

- GNU é a sigla para um nome curioso: "***GNU is Not Unix (GNU Não é Unix)***". Era um projeto que tentava criar um SO Unix, criado em 1984.
- Com o passar dos anos, o projeto foi ganhando recursos, como compiladores e editores de texto. Mas, faltava um elemento importantíssimo: **Um kernel.**
- O Linux que temos hoje é formado por um kernel (Linux) e um conjunto de software GNU.

GNU – Core Utilities

- **GNU Core Utilities** são os utilitários básicos para manipulação de arquivos (*copiar, mover, renomear, excluir, visualizar, localizar*), do shell (*como o nice, pwd, who, chmod, etc*) do sistema operacional GNU.
- **GNU Core Utilities** são os principais utilitários que devem existir em todos os sistemas operacionais, para dar suporte as operações básicas de manipulação do S.O. (neste caso GNU/Linux).

Core Utilities – Porque Compilar?

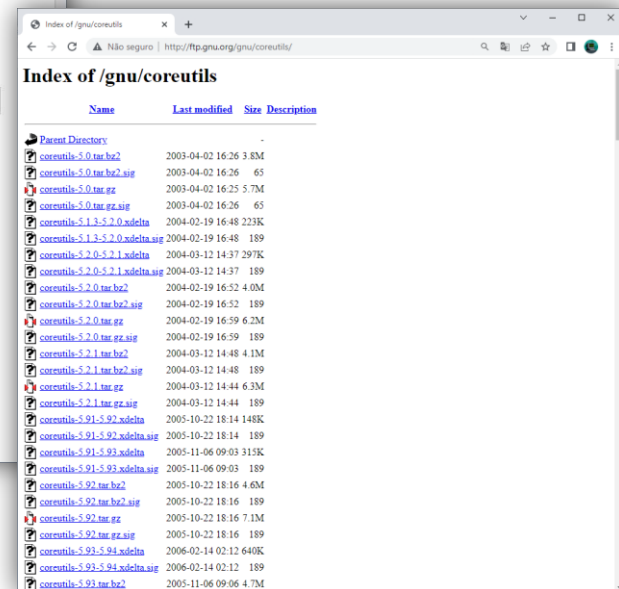
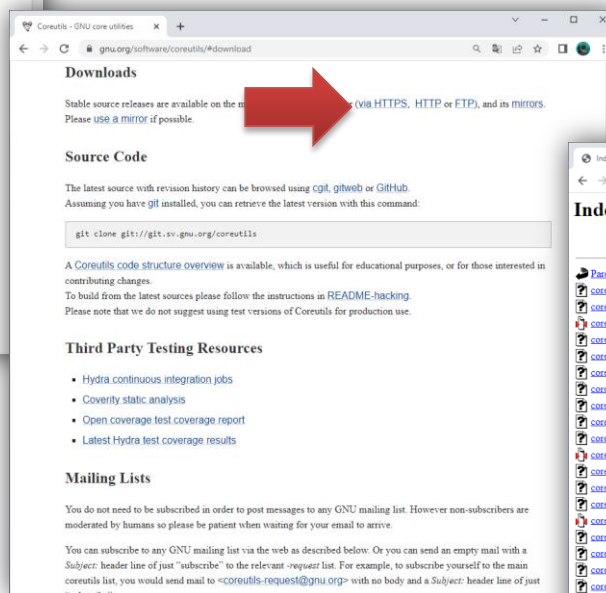
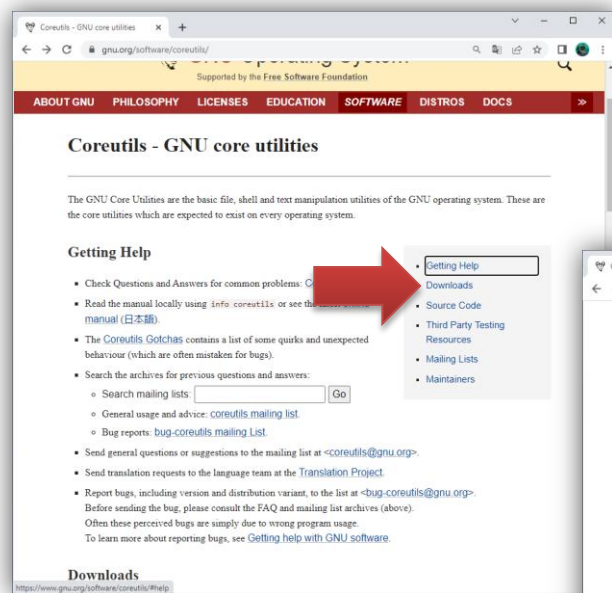
- Necessidade de uma versão específica.
- Criar versão otimizada para hardware específico.
- Melhorias não distribuídas na versão atual.
- Otimização/Customização do Binário.
- Testar melhorias/corrigir falhas.
- Aprendizado.



GNU Core Utilities

Onde baixar os Fontes?

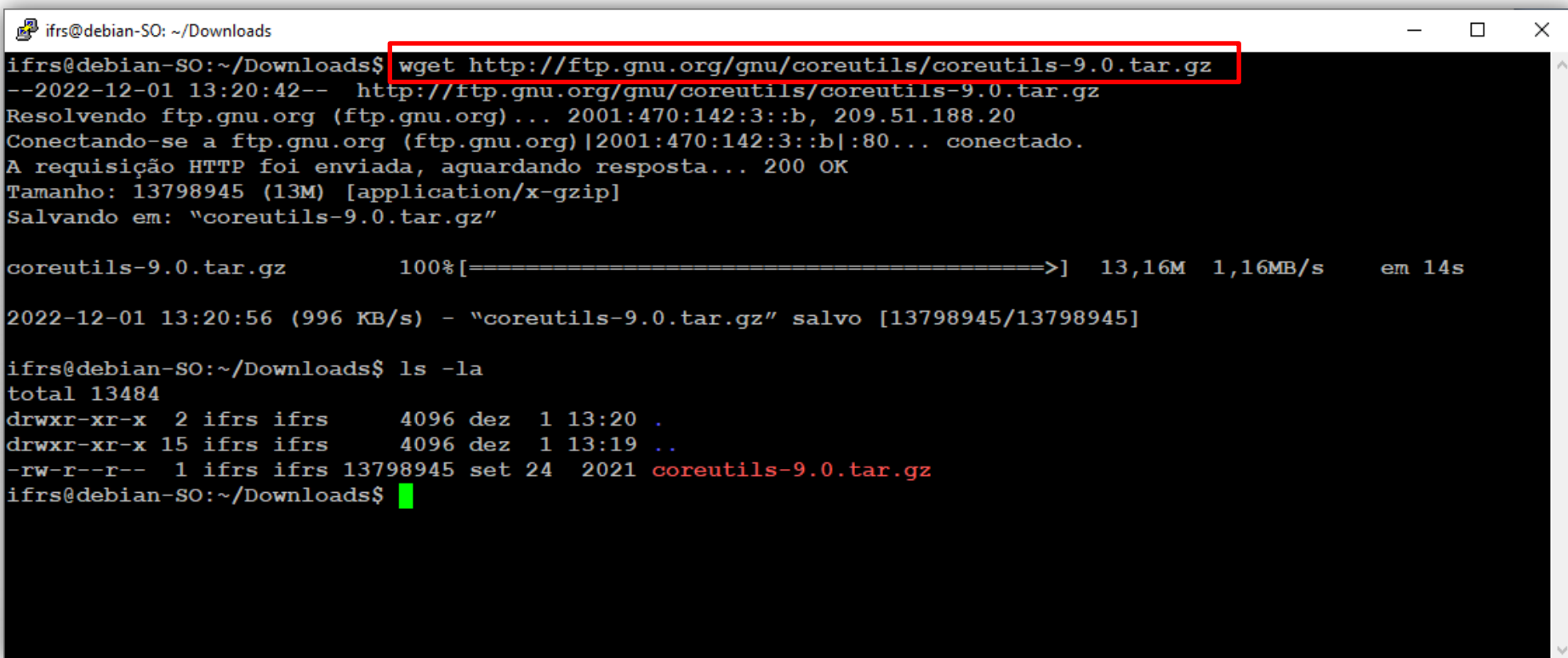
<https://www.gnu.org/software/coreutils/>



Baixando o Fonte coreutils

```
$wget http://ftp.gnu.org/gnu/coreutils/coreutils-9.0.tar.gz
```

wget – utilitário de linha de comando que transfere arquivos.



```
ifrs@debian-SO: ~/Downloads
ifrs@debian-SO:~/Downloads$ wget http://ftp.gnu.org/gnu/coreutils/coreutils-9.0.tar.gz
--2022-12-01 13:20:42-- http://ftp.gnu.org/gnu/coreutils/coreutils-9.0.tar.gz
Resolvendo ftp.gnu.org (ftp.gnu.org)... 2001:470:142:3::b, 209.51.188.20
Conectando-se a ftp.gnu.org (ftp.gnu.org)|2001:470:142:3::b|:80... conectado.
A requisição HTTP foi enviada, aguardando resposta... 200 OK
Tamanho: 13798945 (13M) [application/x-gzip]
Salvando em: "coreutils-9.0.tar.gz"

coreutils-9.0.tar.gz      100%[=====>]  13,16M  1,16MB/s   em 14s

2022-12-01 13:20:56 (996 KB/s) - "coreutils-9.0.tar.gz" salvo [13798945/13798945]

ifrs@debian-SO:~/Downloads$ ls -la
total 13484
drwxr-xr-x  2 ifrs ifrs   4096 dez  1 13:20 .
drwxr-xr-x 15 ifrs ifrs   4096 dez  1 13:19 ..
-rw-r--r--  1 ifrs ifrs 13798945 set 24  2021 coreutils-9.0.tar.gz
ifrs@debian-SO:~/Downloads$
```



Descompactando o Fonte coreutils

```
$ tar xvzf coreutils-9.0.tar.gz -C ./
```

tar – programa para armazenar (empacotar) e extrair arquivos.

x: extrair, **v**: interativo, **z**: compactação GZIP, **f**: especifica o arquivo, **-C** destino

```
ifrs@debian-SO: ~/Downloads
coreutils-9.0/src/die.h
coreutils-9.0/src/head.c
coreutils-9.0/src/ioblksize.h
coreutils-9.0/src/link.c
coreutils-9.0/src/cut.c
coreutils-9.0/src/hostid.c
coreutils-9.0/src/expand-common.c
coreutils-9.0/src/numfmt.c
coreutils-9.0/src/csplit.c
coreutils-9.0/src/stty.c
coreutils-9.0/src/uname.c
coreutils-9.0/src/mkdir.c
coreutils-9.0/src/ptx.c
coreutils-9.0/src/single-binary.mk
coreutils-9.0/src/fmt.c
coreutils-9.0/src/dircolors.c
coreutils-9.0/src/chmod.c
ifrs@debian-SO:~/Downloads$ ls -la
total 13488
drwxr-xr-x  3 ifrs ifrs    4096 dez  1 13:22 .
drwxr-xr-x 15 ifrs ifrs    4096 dez  1 13:19 ..
drwxr-xr-x 11 ifrs ifrs    4096 set 24 2021 coreutils-9.0
-rw-r--r--  1 ifrs ifrs 13798945 set 24 2021 coreutils-9.0.tar.gz
ifrs@debian-SO:~/Downloads$
```

Instalando Pacotes Utilitários

```
#apt install gcc
```

apt : instala pacotes. (como superusuário). **gcc**: utilitários para compilação de .C

```
ifrs@debian-SO: ~/Downloads/coreutils-9.0
ifrs@debian-SO:~/Downloads/coreutils-9.0$ su -
Senha:
root@debian-SO:~# sudo apt install gcc
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
gcc is already the newest version (4:10.2.1-1).
0 pacotes atualizados, 0 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
root@debian-SO:~#
```

Instalando Pacotes Utilitários

```
# apt install git build-essential
```

apt : instala pacotes git e build-essential

```
ifrs@debian-SO: ~/Downloads/coreutils-9.0
root@debian-SO:~# sudo apt install git build-essential
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
build-essential is already the newest version (12.9).
The following additional packages will be installed:
  git-man liberror-perl
Pacotes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki
  git-svn
Os NOVOS pacotes a seguir serão instalados:
  git git-man liberror-perl
0 pacotes atualizados, 3 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 7.386 kB de arquivos.
Depois desta operação, 37,9 MB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] s
Obter:1 http://deb.debian.org/debian bullseye/main amd64 liberror-perl all 0.17029-1 [31,0 kB]
Obter:2 http://deb.debian.org/debian bullseye/main amd64 git-man all 1:2.30.2-1 [1.827 kB]
Obter:3 http://deb.debian.org/debian bullseye/main amd64 git amd64 1:2.30.2-1 [5.527 kB]
Baixados 7.386 kB em 1s (8.948 kB/s)
A seleccionar pacote anteriormente não seleccionado liberror-perl.
(Lendo banco de dados ... 127635 ficheiros e directórios actualmente instalados.)
A preparar para desempacotar .../liberror-perl_0.17029-1_all.deb ...
A descompactar liberror-perl (0.17029-1) ...
```

Instalando Pacotes Utilitários

```
# apt build-dep coreutils
```

Instala (resolve) as dependências (bibliotecas faltantes) para compilação.

```
ifrs@debian-SO: ~/Downloads/coreutils-9.0
root@debian-SO:~# sudo apt build-dep coreutils
Lendo listas de pacotes... Pronto
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
Os NOVOS pacotes a seguir serão instalados:
 autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-strip-nondeterminism dwz gettext
 gperf groff intltool-debian libacl1-dev libarchive-zip-perl libattr1-dev libdebhelper-perl
 libfile-stripnondeterminism-perl libgmp-dev libgmp3-dev libgmpxx4ldbl libpcre2-16-0 libpcre2-32-0
 libpcre2-dev libpcre2-posix2 libselenium1-dev libsepol1-dev libsub-override-perl libtext-unidecode-perl
 libtool libxml-libxml-perl libxml-namespacesupport-perl libxml-sax-base-perl libxml-sax-perl po-debconf
 tex-common texinfo
0 pacotes atualizados, 36 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 15,1 MB de arquivos.
Depois desta operação, 51,9 MB adicionais de espaço em disco serão usados.
Você quer continuar? [S/n] s
Obter:1 http://deb.debian.org/debian bullseye/main amd64 tex-common all 6.16 [53,7 kB]
Obter:2 http://deb.debian.org/debian bullseye/main amd64 autoconf all 2.69-14 [313 kB]
Obter:3 http://deb.debian.org/debian bullseye/main amd64 autotools-dev all 20180224.1+nmu1 [77,1 kB]
Obter:4 http://deb.debian.org/debian bullseye/main amd64 automake all 1:1.16.3-2 [814 kB]
Obter:5 http://deb.debian.org/debian bullseye/main amd64 autopoint all 0.21-4 [510 kB]
Obter:6 http://deb.debian.org/debian bullseye/main amd64 libdebhelper-perl all 13.3.4 [189 kB]
Obter:7 http://deb.debian.org/debian bullseye/main amd64 libtool all 2.4.6-15 [513 kB]
Obter:8 http://deb.debian.org/debian bullseye/main amd64 dh-autoreconf all 20 [17,1 kB]
```




Gerando o Script de Compilação

```
$ ./configure
```

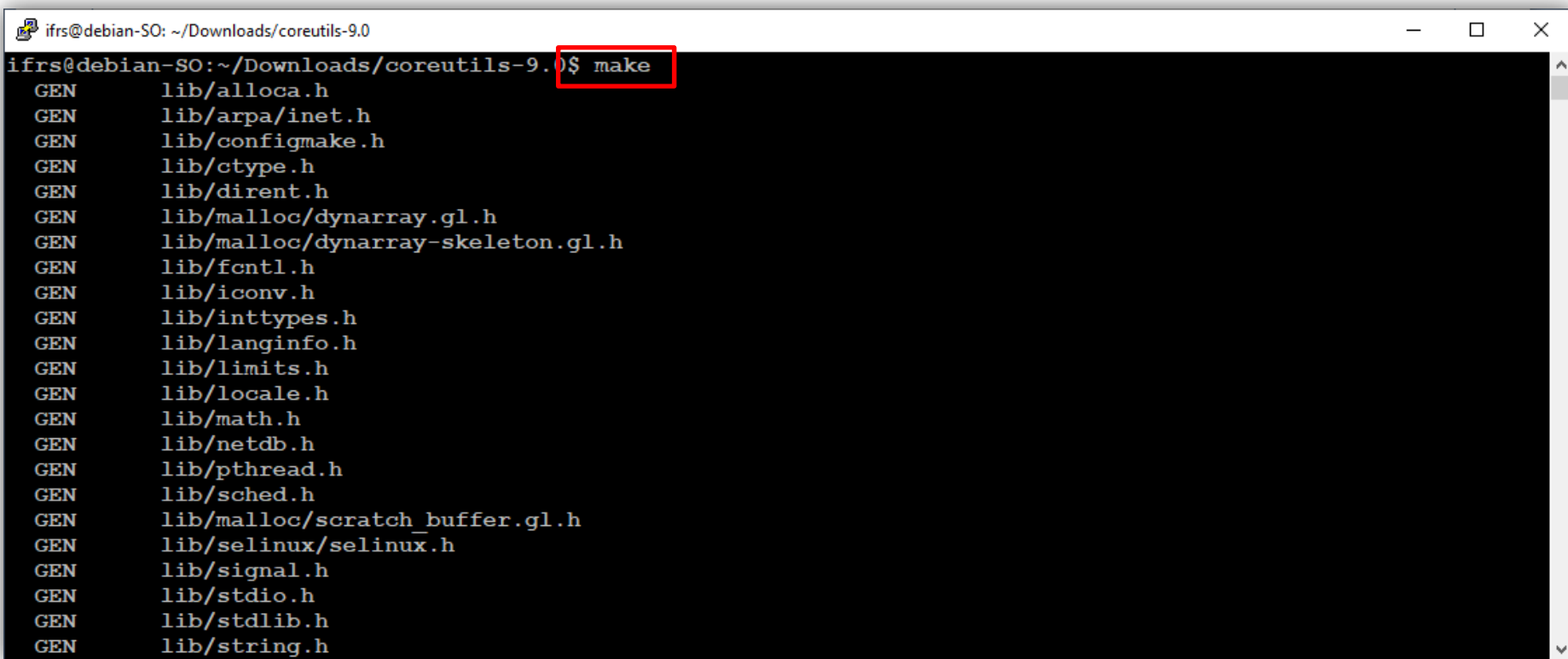
Gera o script de compilação.

```
ifrs@debian-SO: ~/Downloads/coreutils-9.0$ ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether make supports nested variables... (cached) yes
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether gcc accepts -g... yes
checking for gcc option to enable C11 features... none needed
checking whether gcc understands -c and -o together... yes
checking whether the compiler is clang... no
checking for compiler option needed when checking for declarations... none
checking whether make supports the include directive... yes (GNU style)
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
```


Compilação

\$ make

Lê o arquivo gerado (no comando configure) e compila todos os fontes e bibliotecas.



```
ifrs@debian-SO: ~/Downloads/coreutils-9.0$ make
GEN      lib/alloca.h
GEN      lib/arpa/inet.h
GEN      lib/configmake.h
GEN      lib/ctype.h
GEN      lib/dirent.h
GEN      lib/malloc/dynarray.gl.h
GEN      lib/malloc/dynarray-skeleton.gl.h
GEN      lib/fcntl.h
GEN      lib/iconv.h
GEN      lib/inttypes.h
GEN      lib/langinfo.h
GEN      lib/limits.h
GEN      lib/locale.h
GEN      lib/math.h
GEN      lib/netdb.h
GEN      lib/pthread.h
GEN      lib/sched.h
GEN      lib/malloc/scratch_buffer.gl.h
GEN      lib/selinux/selinux.h
GEN      lib/signal.h
GEN      lib/stdio.h
GEN      lib/stdlib.h
GEN      lib/string.h
```

Testando

```
$cd src
```

```
$ls -version (versão instalada no /bin)
```

```
$/ls -version (nova versão compilada)
```

```
ifrs@debian-SO: ~/Downloads/coreutils-9.0/src
ifrs@debian-SO:~/Downloads/coreutils-9.0$ cd src
ifrs@debian-SO:~/Downloads/coreutils-9.0/src$ ls --version
ls (GNU coreutils) 8.32
Copyright (C) 2020 Free Software Foundation, Inc.
Licença GPLv3+: GNU GPL versão 3 ou posterior <https://gnu.org/licenses/gpl.html>
Este é um software livre: você é livre para alterá-lo e redistribuí-lo.
NÃO HÁ QUALQUER GARANTIA, na máxima extensão permitida em lei.

Escrito por Richard M. Stallman e David MacKenzie.
ifrs@debian-SO:~/Downloads/coreutils-9.0/src$ ./ls --version
ls (GNU coreutils) 9.0
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Richard M. Stallman and David MacKenzie.
ifrs@debian-SO:~/Downloads/coreutils-9.0/src$
```