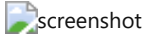


Apenas o [README](#) original é garantido estar atualizado.

scrcpy (v1.19)

Esta aplicação fornece exibição e controle de dispositivos Android conectados via USB (ou [via TCP/IP](#)). Não requer nenhum acesso *root*. Funciona em *GNU/Linux*, *Windows* e *macOS*.



screenshot

Foco em:

- **leveza** (nativo, mostra apenas a tela do dispositivo)
- **performance** (30~60fps)
- **qualidade** (1920×1080 ou acima)
- **baixa latência** ([35~70ms](#))
- **baixo tempo de inicialização** (~1 segundo para mostrar a primeira imagem)
- **não intrusivo** (nada é deixado instalado no dispositivo)

Requisitos

O dispositivo Android requer pelo menos a API 21 (Android 5.0).

Tenha certeza de ter [ativado a depuração adb](#) no(s) seu(s) dispositivo(s).

Em alguns dispositivos, você também precisa ativar [uma opção adicional](#) para controlá-lo usando teclado e mouse.

Obter o app

Sumário

- Linux: `apt install scrcpy`
- Windows: [baixar][direct-win64]
- macOS: `brew install scrcpy`

Compilar pelos arquivos fontes: [BUILD](#) ([processo simplificado](#))

Linux

No Debian (*testing* e *sid* por enquanto) e Ubuntu (20.04):

```
apt install scrcpy
```

Um pacote [Snap](#) está disponível: [scrcpy](#) .

Para Fedora, um pacote [COPR](#) está disponível: [scrcpy](#) .

Para Arch Linux, um pacote [AUR](#) está disponível: [scrcpy](#) .

Para Gentoo, uma [Ebuild](#) está disponível: [scrcpy/](#) .

Você também pode [compilar o app manualmente](#) ([processo simplificado](#)).

Windows

Para Windows, por simplicidade, um arquivo pré-compilado com todas as dependências (incluindo `adb`) está disponível:

- [README](#)

Também está disponível em [Chocolatey](#):

```
choco install srcpy
choco install adb      # se você ainda não o tem
```

E no [Scoop](#):

```
scoop install srcpy
scoop install adb      # se você ainda não o tem
```

Você também pode [compilar o app manualmente](#).

macOS

A aplicação está disponível em [Homebrew](#). Apenas instale-a:

```
brew install srcpy
```

Você precisa do `adb` , acessível pelo seu `PATH` . Se você ainda não o tem:

```
brew install android-platform-tools
```

Está também disponível em [MacPorts](#), que prepara o adb para você:

```
sudo port install srcpy
```

Você também pode [compilar o app manualmente](#).

Executar

Conecte um dispositivo Android e execute:

```
srcpy
```

Também aceita argumentos de linha de comando, listados por:

```
srcpy --help
```

Funcionalidades

Configuração de captura

Packaging status	
Alpine Linux 3.16	1.24
Alpine Linux Edge	1.24
ALT Linux p9	1.16
ALT Linux p10	1.21
ALT Sisyphus	1.21
antiX-19	1.12.1
AOSC	1.24
Arch	1.24
Arch Linux 32 i686	1.24
Arch Linux 32 pentium4	1.24
Arch Linux ARM aarch64	1.24
AUR	1.17.r3.ge...
Chocolatey	1.24
Debian 11	1.17
Debian 11 Backports	1.23
Debian 12	1.24
Debian Unstable	1.24
Devuan 4.0	1.17
Devuan Unstable	1.24
DPorts	1.9
FreeBSD Ports	1.24
Funtoo 1.4	1.24
Gentoo	1.24
Homebrew	1.24
Kali Linux Rolling	1.24
LiGurOS stable	1.24
LiGurOS develop	1.24
MacPorts	1.24
Manjaro Stable	1.24
Manjaro Testing	1.24
Manjaro Unstable	1.24
MPR	1.24
MSYS2 mingw	1.24
MX Linux MX-17	1.12.1
MX Linux MX-19	1.12.1
nixpkgs stable 21.05	1.17
nixpkgs stable 21.11	1.20
nixpkgs stable 22.05	1.24
nixpkgs unstable	1.24
OpenMandriva 4.1	1.12.1
OpenMandriva 4.2	1.17
OpenMandriva Rolling	1.24
OpenMandriva Cooker	1.24

Reduzir tamanho

Algumas vezes, é útil espelhar um dispositivo Android em uma resolução menor para aumentar a performance.

Para limitar ambos (largura e altura) para algum valor (ex: 1024):

```
scrcpy --max-size 1024
scrcpy -m 1024 # versão curta
```

A outra dimensão é calculada para que a proporção do dispositivo seja preservada. Dessa forma, um dispositivo de 1920x1080 será espelhado em 1024x576.

Mudar bit-rate

O bit-rate padrão é 8 Mbps. Para mudar o bit-rate do vídeo (ex: para 2 Mbps):

```
scrcpy --bit-rate 2M
scrcpy -b 2M # versão curta
```

Limitar frame rate

O frame rate de captura pode ser limitado:

```
scrcpy --max-fps 15
```

Isso é oficialmente suportado desde o Android 10, mas pode funcionar em versões anteriores.

Cortar

A tela do dispositivo pode ser cortada para espelhar apenas uma parte da tela.

Isso é útil por exemplo, para espelhar apenas um olho do Oculus Go:

```
scrcpy --crop 1224:1440:0:0 # 1224x1440 no deslocamento (0,0)
```

Se `--max-size` também for especificado, o redimensionamento é aplicado após o corte.

Travar orientação do vídeo

Para travar a orientação do espelhamento:

```
scrcpy --lock-video-orientation # orientação inicial (Atual)
scrcpy --lock-video-orientation=0 # orientação natural
scrcpy --lock-video-orientation=1 # 90° sentido anti-horário
scrcpy --lock-video-orientation=2 # 180°
scrcpy --lock-video-orientation=3 # 90° sentido horário
```

Isso afeta a orientação de gravação.

Parabola	1.24
Pardus 21	1.17
Parrot	1.17
Pisi Linux	1.24
PureOS landing	1.17
Raspbian Stable	1.17
Raspbian Testing	1.24
RPM Sphere	1.24
Scoop	1.24
SlackBuilds	1.24
Solus	1.24
Trisquel 10.0	1.12.1
Ubuntu 20.04	1.12.1
Ubuntu 22.04	1.21
Ubuntu 22.10	1.24
Void Linux x86_64	1.24

A [janela também pode ser rotacionada](#) independentemente.

Encoder

Alguns dispositivos têm mais de um encoder, e alguns deles podem causar problemas ou travar. É possível selecionar um encoder diferente:

```
scrcpy --encoder OMX.qcom.video.encoder.avc
```

Para listar os encoders disponíveis, você pode passar um nome de encoder inválido, o erro dará os encoders disponíveis:

```
scrcpy --encoder _
```

Captura

Gravando

É possível gravar a tela enquanto ocorre o espelhamento:

```
scrcpy --record file.mp4
scrcpy -r file.mkv
```

Para desativar o espelhamento durante a gravação:

```
scrcpy --no-display --record file.mp4
scrcpy -Nr file.mkv
# interrompa a gravação com Ctrl+C
```

"Frames pulados" são gravados, mesmo que não sejam exibidos em tempo real (por motivos de performance). Frames têm seu *horário carimbado* no dispositivo, então [variação de atraso nos pacotes](#) não impacta o arquivo gravado.

v4l2loopback

Em Linux, é possível enviar a transmissão do vídeo para um dispositivo v4l2 loopback, assim o dispositivo Android pode ser aberto como uma webcam por qualquer ferramenta capaz de v4l2

The module `v4l2loopback` must be installed:

```
sudo apt install v4l2loopback-dkms
```

Para criar um dispositivo v4l2:

```
sudo modprobe v4l2loopback
```

Isso cria um novo dispositivo de vídeo em `/dev/videoN`, onde `N` é uma integer (mais [opções](#) estão disponíveis para criar vários dispositivos ou dispositivos com IDs específicas).

Para listar os dispositivos disponíveis:

```
# requer o pacote v4l-utils
v4l2-ctl --list-devices

# simples, mas pode ser suficiente
ls /dev/video*
```

Para iniciar o scrcpy usando o coletor v4l2 (sink):

```
scrcpy --v4l2-sink=/dev/videoN
scrcpy --v4l2-sink=/dev/videoN --no-display # desativa a janela espelhada
scrcpy --v4l2-sink=/dev/videoN -N          # versão curta
```

(troque `N` pelo ID do dispositivo, verifique com `ls /dev/video*`)

Uma vez ativado, você pode abrir suas transmissões de vídeos com uma ferramenta capaz de v4l2:

```
ffplay -i /dev/videoN
vlc v4l2:///dev/videoN # VLC pode adicionar um pouco de atraso de buffering
```

Por exemplo, você pode capturar o vídeo dentro do [OBS](#).

Buffering

É possível adicionar buffering. Isso aumenta a latência, mas reduz a tensão (jitter) (veja [#2464](#)).

A opção é disponível para buffering de exibição:

```
scrcpy --display-buffer=50 # adiciona 50 ms de buffering para a exibição
```

e coletor V4L2:

```
scrcpy --v4l2-buffer=500 # adiciona 500 ms de buffering para coletor V4L2
```

Conexão

Sem fio

Scrcpy usa `adb` para se comunicar com o dispositivo, e `adb` pode [conectar-se](#) a um dispositivo via TCP/IP:

1. Conecte o dispositivo no mesmo Wi-Fi do seu computador.
2. Pegue o endereço IP do seu dispositivo, em Configurações → Sobre o telefone → Status, ou executando este comando:

```
adb shell ip route | awk '{print $9}'
```

3. Ative o adb via TCP/IP no seu dispositivo: `adb tcpip 5555` .
4. Desconecte seu dispositivo.

5. Conecte-se ao seu dispositivo: `adb connect DEVICE_IP:5555` (*substitua* `DEVICE_IP`).

6. Execute `scrcpy` como de costume.

Pode ser útil diminuir o bit-rate e a resolução:

```
scrcpy --bit-rate 2M --max-size 800  
scrcpy -b2M -m800 # versão curta
```

Múltiplos dispositivos

Se vários dispositivos são listados em `adb devices`, você deve especificar o *serial*:

```
scrcpy --serial 0123456789abcdef  
scrcpy -s 0123456789abcdef # versão curta
```

Se o dispositivo está conectado via TCP/IP:

```
scrcpy --serial 192.168.0.1:5555  
scrcpy -s 192.168.0.1:5555 # versão curta
```

Você pode iniciar várias instâncias do *scrcpy* para vários dispositivos.

Iniciar automaticamente quando dispositivo é conectado

Você pode usar [AutoAdb](#):

```
autoadb scrcpy -s '{}'
```

Túnel SSH

Para conectar-se a um dispositivo remoto, é possível conectar um cliente `adb` local a um servidor `adb` remoto (contanto que eles usem a mesma versão do protocolo *adb*):

```
adb kill-server # encerra o servidor adb local em 5037  
ssh -CN -L5037:localhost:5037 -R27183:localhost:27183 your_remote_computer  
# mantenha isso aberto
```

De outro terminal:

```
scrcpy
```

Para evitar ativar o encaminhamento de porta remota, você pode forçar uma conexão de encaminhamento (note o `-L` em vez de `-R`):

```
adb kill-server # encerra o servidor adb local em 5037  
ssh -CN -L5037:localhost:5037 -L27183:localhost:27183 your_remote_computer  
# mantenha isso aberto
```

De outro terminal:

```
scrcpy --force-adb-forward
```

Igual a conexões sem fio, pode ser útil reduzir a qualidade:

```
scrcpy -b2M -m800 --max-fps 15
```

Configuração de janela

Título

Por padrão, o título da janela é o modelo do dispositivo. Isso pode ser mudado:

```
scrcpy --window-title 'Meu dispositivo'
```

Posição e tamanho

A posição e tamanho iniciais da janela podem ser especificados:

```
scrcpy --window-x 100 --window-y 100 --window-width 800 --window-height 600
```

Sem bordas

Para desativar decorações de janela:

```
scrcpy --window-borderless
```

Sempre no topo

Para manter a janela do scrcpy sempre no topo:

```
scrcpy --always-on-top
```

Tela cheia

A aplicação pode ser iniciada diretamente em tela cheia:

```
scrcpy --fullscreen  
scrcpy -f # versão curta
```

Tela cheia pode ser alternada dinamicamente com `MOD+F`.

Rotação

A janela pode ser rotacionada:

```
scrcpy --rotation 1
```

Valores possíveis são:

- 0 : sem rotação
- 1 : 90 graus sentido anti-horário
- 2 : 180 graus
- 3 : 90 graus sentido horário

A rotação também pode ser mudada dinamicamente com `MOD+←` (*esquerda*) e `MOD+→` (*direita*).

Note que `scrcpy` controla 3 rotações diferentes:

- `MOD+r` requisita ao dispositivo para mudar entre retrato e paisagem (a aplicação em execução pode se recusar, se ela não suporta a orientação requisitada).
- `--lock-video-orientation` muda a orientação de espelhamento (a orientação do vídeo enviado pelo dispositivo para o computador). Isso afeta a gravação.
- `--rotation` (ou `MOD+←/MOD+→`) rotaciona apenas o conteúdo da janela. Isso afeta apenas a exibição, não a gravação.

Outras opções de espelhamento

Apenas leitura

Para desativar controles (tudo que possa interagir com o dispositivo: teclas de entrada, eventos de mouse, arrastar e soltar arquivos):

```
scrcpy --no-control
scrcpy -n
```

Display

Se vários displays estão disponíveis, é possível selecionar o display para espelhar:

```
scrcpy --display 1
```

A lista de IDs dos displays pode ser obtida por:

```
adb shell dumpsys display # busca "mDisplayId=" na saída
```

O display secundário pode apenas ser controlado se o dispositivo roda pelo menos Android 10 (caso contrário é espelhado como apenas leitura).

Permanecer ativo

Para evitar que o dispositivo seja suspenso após um delay quando o dispositivo é conectado:

```
scrcpy --stay-awake
scrcpy -w
```

O estado inicial é restaurado quando o `scrcpy` é fechado.

Desligar tela

É possível desligar a tela do dispositivo durante o início do espelhamento com uma opção de linha de comando:


```
scrcpy --turn-screen-off
scrcpy -S
```

Ou apertando `MOD+o` a qualquer momento.

Para ligar novamente, pressione `MOD+Shift+o`.

No Android, o botão de `POWER` sempre liga a tela. Por conveniência, se `POWER` é enviado via `scrcpy` (via clique-direito ou `MOD+p`), ele forçará a desligar a tela após um delay pequeno (numa base de melhor esforço). O botão `POWER` físico ainda causará a tela ser ligada.

Também pode ser útil evitar que o dispositivo seja suspenso:

```
scrcpy --turn-screen-off --stay-awake
scrcpy -Sw
```

Mostrar toques

Para apresentações, pode ser útil mostrar toques físicos (no dispositivo físico).

Android fornece esta funcionalidade nas *Opções do desenvolvedor*.

Scrcpy fornece esta opção de ativar esta funcionalidade no início e restaurar o valor inicial no encerramento:

```
scrcpy --show-touches
scrcpy -t
```

Note que isto mostra apenas toques *físicos* (com o dedo no dispositivo).

Desativar descanso de tela

Por padrão, `scrcpy` não evita que o descanso de tela rode no computador.

Para desativá-lo:

```
scrcpy --disable-screensaver
```

Controle de entrada

Rotacionar a tela do dispositivo

Pressione `MOD+r` para mudar entre os modos retrato e paisagem.

Note que só será rotacionado se a aplicação em primeiro plano suportar a orientação requisitada.

Copiar-colar

Sempre que a área de transferência do Android muda, é automaticamente sincronizada com a área de transferência do computador.

Qualquer atalho com `Ctrl` é encaminhado para o dispositivo. Em particular:

- `Ctrl+c` tipicamente copia
- `Ctrl+x` tipicamente recorta

- `Ctrl+v` tipicamente cola (após a sincronização de área de transferência computador-para-dispositivo)

Isso tipicamente funciona como esperado.

O comportamento de fato depende da aplicação ativa, no entanto. Por exemplo, *Termux* envia SIGINT com `Ctrl+c`, e *K-9 Mail* compõe uma nova mensagem.

Para copiar, recortar e colar em tais casos (mas apenas suportado no Android ≥ 7):

- `MOD+c` injeta `COPY`
- `MOD+x` injeta `CUT`
- `MOD+v` injeta `PASTE` (após a sincronização de área de transferência computador-para-dispositivo)

Em adição, `MOD+Shift+v` permite injetar o texto da área de transferência do computador como uma sequência de eventos de tecla. Isso é útil quando o componente não aceita colar texto (por exemplo no *Termux*), mas pode quebrar conteúdo não-ASCII.

ADVERTÊNCIA: Colar a área de transferência do computador para o dispositivo (tanto via `Ctrl+v` quanto `MOD+v`) copia o conteúdo para a área de transferência do dispositivo. Como consequência, qualquer aplicação Android pode ler o seu conteúdo. Você deve evitar colar conteúdo sensível (como senhas) dessa forma.

Alguns dispositivos não se comportam como esperado quando a área de transferência é definida programaticamente. Uma opção `--legacy-paste` é fornecida para mudar o comportamento de `Ctrl+v` e `MOD+v` para que eles também injetem o texto da área de transferência do computador como uma sequência de eventos de tecla (da mesma forma que `MOD+Shift+v`).

Pinçar para dar zoom

Para simular "pinçar para dar zoom": `Ctrl+clicar-e-mover`.

Mais precisamente, segure `Ctrl` enquanto pressiona o botão de clique-esquerdo. Até que o botão de clique-esquerdo seja liberado, todos os movimentos do mouse ampliar e rotacionam o conteúdo (se suportado pelo app) relativo ao centro da tela.

Concretamente, *scrcpy* gera eventos adicionais de toque de um "dedo virtual" em uma posição invertida em relação ao centro da tela.

Preferência de injeção de texto

Existem dois tipos de [eventos](#) gerados ao digitar um texto:

- *eventos de tecla*, sinalizando que a tecla foi pressionada ou solta;
- *eventos de texto*, sinalizando que o texto foi inserido.

Por padrão, letras são injetadas usando eventos de tecla, assim o teclado comporta-se como esperado em jogos (normalmente para teclas WASD).

Mas isso pode [causar problemas](#). Se você encontrar tal problema, você pode evitá-lo com:

```
scrcpy --prefer-text
```

(mas isso vai quebrar o comportamento do teclado em jogos)

Repetir tecla

Por padrão, segurar uma tecla gera eventos de tecla repetidos. Isso pode causar problemas de performance em alguns jogos, onde esses eventos são inúteis de qualquer forma.

Para evitar o encaminhamento eventos de tecla repetidos:

```
scrcpy --no-key-repeat
```

Clique-direito e clique-do-meio

Por padrão, clique-direito dispara BACK (ou POWER) e clique-do-meio dispara HOME. Para desabilitar esses atalhos e encaminhar os cliques para o dispositivo:

```
scrcpy --forward-all-clicks
```

Soltar arquivo

Instalar APK

Para instalar um APK, arraste e solte o arquivo APK (com extensão `.apk`) na janela *scrcpy*.

Não existe feedback visual, um log é imprimido no console.

Enviar arquivo para dispositivo

Para enviar um arquivo para `/sdcard/Download/` no dispositivo, arraste e solte um arquivo (não-APK) para a janela do *scrcpy*.

Não existe feedback visual, um log é imprimido no console.

O diretório alvo pode ser mudado ao iniciar:

```
scrcpy --push-target /sdcard/foo/bar/
```

Encaminhamento de áudio

Áudio não é encaminhado pelo *scrcpy*. Use [sndcpy](#).

Também veja [issue #14](#).

Atalhos

Na lista a seguir, `MOD` é o modificador de atalho. Por padrão, é `Alt` (esquerdo) ou `Super` (esquerdo).

Ele pode ser mudado usando `--shortcut-mod`. Possíveis teclas são `lctrl`, `rctrl`, `lalt`, `ralt`, `lsuper` e `rsuper`. Por exemplo:

```
# usar RCtrl para atalhos
scrcpy --shortcut-mod=rctrl

# usar tanto LCtrl+LAlt quanto LSuper para atalhos
scrcpy --shortcut-mod=lctrl+lalt,lsuper
```

[Super](#) é tipicamente a tecla *Windows* ou *Cmd*.

Ação	Atalho

Mudar modo de tela cheia	MOD+f
Rotacionar display para esquerda	MOD+← (<i>esquerda</i>)
Rotacionar display para direita	MOD+→ (<i>direita</i>)
Redimensionar janela para 1:1 (pixel-perfeito)	MOD+g
Redimensionar janela para remover bordas pretas	MOD+w <i>Clique-duplo-esquerdo</i> ¹
Clicar em HOME	MOD+h <i>Clique-do-meio</i>
Clicar em BACK	MOD+b <i>Clique-direito</i> ²
Clicar em APP_SWITCH	MOD+s <i>Clique-do-4.</i> ^{°3}
Clicar em MENU (desbloquear tela)	MOD+m
Clicar em VOLUME_UP	MOD+↑ (<i>cima</i>)
Clicar em VOLUME_DOWN	MOD+↓ (<i>baixo</i>)
Clicar em POWER	MOD+p
Ligar	<i>Clique-direito</i> ²
Desligar tela do dispositivo (continuar espelhando)	MOD+o
Ligar tela do dispositivo	MOD+Shift+o
Rotacionar tela do dispositivo	MOD+r
Expandir painel de notificação	MOD+n <i>Clique-do-5.</i> ^{°3}
Expandir painel de configuração	MOD+n+n <i>Clique-duplo-do-5.</i> ^{°3}
Colapsar painéis	MOD+Shift+n
Copiar para área de transferência ⁴	MOD+c
Recortar para área de transferência ⁴	MOD+x
Sincronizar áreas de transferência e colar ⁴	MOD+v
Injetar texto da área de transferência do computador	MOD+Shift+v
Ativar/desativar contador de FPS (em stdout)	MOD+i
Pinçar para dar zoom	Ctrl+Clicar-e-mover

¹*Clique-duplo-esquerdo na borda preta para remove-la.*

²*Clique-direito liga a tela caso esteja desligada, pressione BACK caso contrário.*

³*4.° and 5.° botões do mouse, caso o mouse possua.*

⁴*Apenas em Android >= 7.*

Atalhos com teclas reptidas são executados soltando e precionando a tecla uma segunda vez. Por exemplo, para executar "Expandir painel de Configuração":

1. Mantenha pressionado MOD.
2. Depois click duas vezes n.

3. Finalmente, solte `MOD`.

Todos os atalhos `Ctrl+tecla` são encaminhados para o dispositivo, para que eles sejam tratados pela aplicação ativa.

Caminhos personalizados

Para usar um binário `adb` específico, configure seu caminho na variável de ambiente `ADB` :

```
ADB=/caminho/para/adb scrpcy
```

Para sobrepor o caminho do arquivo `scrpcy-server` , configure seu caminho em `SCRCPY_SERVER_PATH` .

Por quê *scrpcy*?

Um colega me desafiou a encontrar um nome tão impronunciável quanto [gnirehtet](#).

[scrpcy](#) copia uma **string**; `scrpcy` copia uma **screen**.

Como compilar?

Veja [BUILD](#).

Problemas comuns

Veja o [FAQ](#).

Desenvolvedores

Leia a [página dos desenvolvedores](#).

Licença

Copyright (C) 2018 Genymobile

Copyright (C) 2018-2022 Romain Vimont

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Artigos

- [Introducing scrpcy](#)

- [Scrcpy now works wirelessly.](#)