

POLITÉCNICO DO PORTO
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Sistema de Vigilância

Stéphane Monteiro, João Dias

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização em Automação e Sistemas



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto

Setembro, 2022

*Esta dissertação satisfaz, parcialmente, os requisitos que constam da Ficha de
Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em
Engenharia Electrotécnica e de Computadores, Área de Especialização em
Automação e Sistemas.*

Candidatos: Stéphane Monteiro, João Dias, N.º 1180697, 1181617,
1180697@isep.ipp.pt, 1181617@isep.ipp.pt

Orientação Científica: Jorge Estrela da Silva, jes@isep.ipp.pt



DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Setembro, 2022

Índice

Lista de Figuras	vii
1 Introdução	1
2 Implementação	3
3 Configurações Buildroot	5
4 Configurações Kernel	11
5 QEMU	15
6 Dispositivo de armazenamento	17
7 Resultados	21
8 Conclusões	25
Referências	26

Lista de Figuras

3.1	Target Options	5
3.2	Toolchain	6
3.3	Toolchain-Continuação	6
3.4	System configuration	7
3.5	Kernel	7
3.6	VLCutils	8
3.7	Ffmpeg	8
3.8	Hardware Handling	8
3.9	Filesystem Images	9
4.1	Multimedia Support	12
4.2	UVC	12
4.3	Câmara Drivers	12
4.4	Graphic Drivers	12
4.5	USB	13
7.1	Arranque do sistema	21
7.2	VLC Network	22
7.3	VLC com stream online	22
7.4	Informação do CPU	23
7.5	RAM utilizada	23
7.6	Disco usada df	23
7.7	Disco usada du	24

Capítulo 1

Introdução

No âmbito da disciplina de Arquitetura de Computadores, do Mestrado em Engenharia Electrotécnica e de Computadores, desenvolveu-se uma distribuição Linux capaz de gravar e reproduzir vídeo e imagem. Adicionalmente acrescentou-se a possibilidade de stream para máquinas dentro da mesma rede LAN. Estas gravações utilizaram uma webcam USB, a Logitech C170.

O estudo inicial baseou-se na pesquisa de um equipamento de gravação, webcam, o modelo c170 da logitech foi a nossa escolha. A pesquisa seguinte consistiu em encontrar os drivers mais indicados para o dispositivo. Para isso seguiu-se o seguinte tutorial referenciado [1][2]. Após estas etapas iniciais foi possível avançar para as configurações do kernel. Ao longo dos ensaios experimentais, denotou-se a necessidade de codecs para capturar e reproduzir imagem e vídeo, respetivamente.

As dificuldades desta implementação encontraram-se sobretudo nas configurações incorretas do kernel. Contudo depois de despistar as mesmas foi possível validar a solução.

Os dispositivos utilizados para os ensaios do sistema de vigilância foram os seguintes:

- Desktop: CPU-Ryzen 1600, RAM-16GB, GPU-NVIDIA GEFORCE 1060 6GB;
- Webcam Logitech C170;
- Dispositivo de armazenamento USB 8GB;

A webcam USB é capaz de capturar vídeo com uma resolução 640x480 e utiliza USB 2.0 como meio de conexão, os drivers utilizados pela mesma são os seguintes [3]:

- OV51X/OVFX2/W996xCF;
- OV534 OV772x;
- OV534 OV965x.

O software para reproduzir os ficheiros capturados, foi o VLC [4]. De modo assegurar uma boa qualidade de imagem, é necessário que o formato de saída do ficheiro e a taxa de frames sejam adequados. Os formatos escolhidos para os ensaios foram o .mkv (Matroska) e .mp4 (MPEG-4), já o codec foi o huffyuv [5] disponível no programa ffmpeg [6]. O ffmpeg é um conversor de vídeo e áudio. Para o stream do vídeo capturado recorreu se a outro codec no programa ffmpeg o mpeg2video.

Capítulo 2

Implementação

Biblioteca

De forma a não existir contratempos durante a compilação do kernel e buildroot devem ser instaladas as seguintes bibliotecas através dos seguintes comandos, denotar que o projeto foi realizado em Fedora 34:

```
$ su
$ dnf install binutils bzip2 unzip gcc make autoconf automake
↪ libtool
$ dnf install patch git subversion bison flex gettext gawk
↪ texinfo gperf
$ dnf install quilt screen
$ dnf install qt-devel ncurses-devel perl perl-ExtUtils*
↪ zlib-devel gcc-c++
$ exit
```

A versão do Buildroot, utilizada no desenvolvimento deste projeto, foi a versão 2021.02.7 [7].

```
$ cd ~
$ mkdir projetoarcom
$ cd projetoarcom
$ wget https://buildroot.org/downloads/buildroot-2021.02.7.tar.bz2
```

```
$ tar xvf buildroot-2021.02.7.tar.bz2
$ cd buildroot-2021.02.7
$ make allnoconfig
$ make pc_x86_64_efi_defconfig
$ make xconfig
```

O comando `make pc_x86_64_efi_defconfig` insere as configurações base de uma arquitetura x86.

Capítulo 3

Configurações Buildroot

É importante referenciar que as configurações que se seguem, estão disponibilizadas nos guiões laboratoriais 3 e 5, [8][9] da unidade curricular de ARCOM.

Em "Target options" verifique que as configurações são iguais às apresentadas na Figura 3.1.

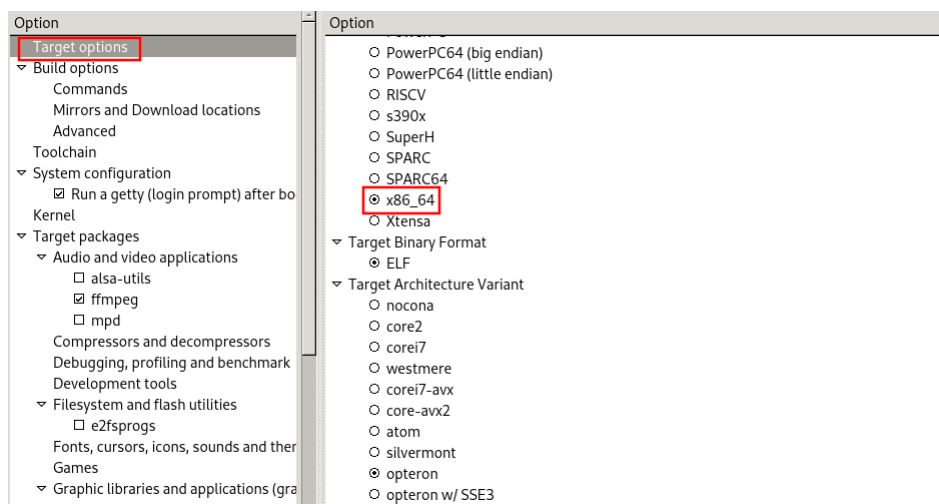


Figura 3.1: Target Options

No "Toolchain":

- Em "C library", verifique que tem a opção "uClibc-ng" selecionada
- Selecione "Enable toolchain locale/i18n support"

- Em "GCC compiler Version", verifique que tem a opção "gcc 9.x" selecionada
- Selecione "Enable C++ support"

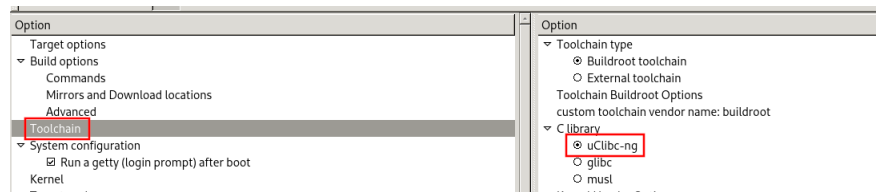


Figura 3.2: Toolchain

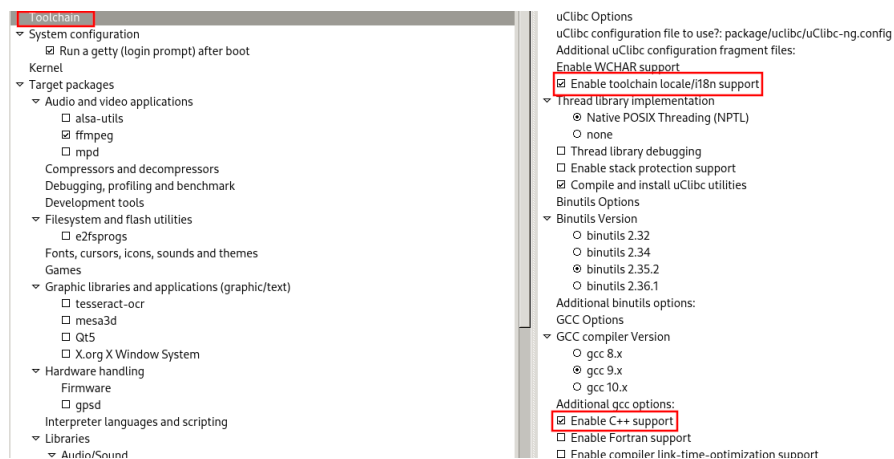


Figura 3.3: Toolchain-Continuação

No "System configuration":

- Altere o "System Banner" para **Sistema de Vigilância**
- Altere o "Password encoding" para "sha-512"
- Em "init system", selecione "busybox"
- Em "Enable root login with password" defina a "Root password" como "root"

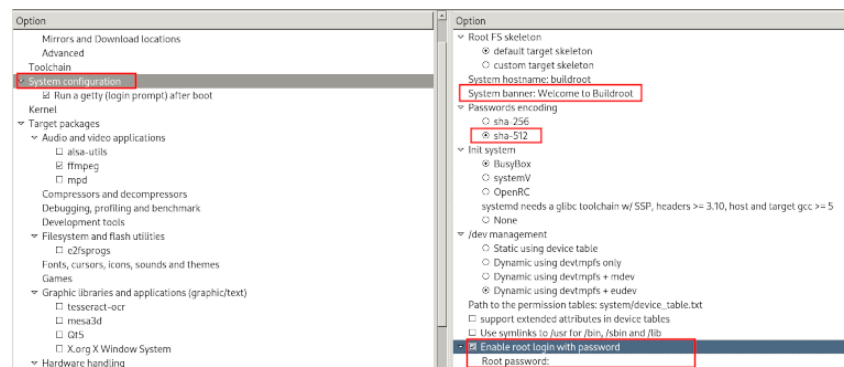


Figura 3.4: System configuration

No "Kernel":

- Em "Kernel version", escolha "Custom version". Depois, clique em "Kernel version" e escreva 4.18.10

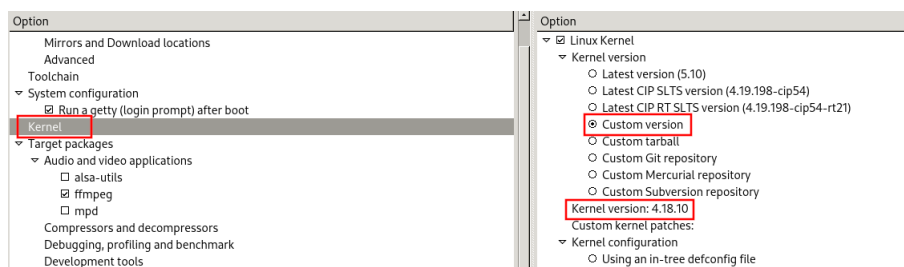


Figura 3.5: Kernel

No "Target packages":

- Em "Audio and video applications" selecione, Figura 3.6:

- "v4l2grab"
- "v4l2loopback" e "utils"
- "vlc"

- Habilite a opção "ffmpeg", Figura 3.7

- Em "ffmpeg" selecione:

- "Build ffmpeg (the command line application)"
- "Build ffplay"

- Em "Libraries->Graphics->Hardware handling" selecione "v4l-utils tools", Figura 3.8

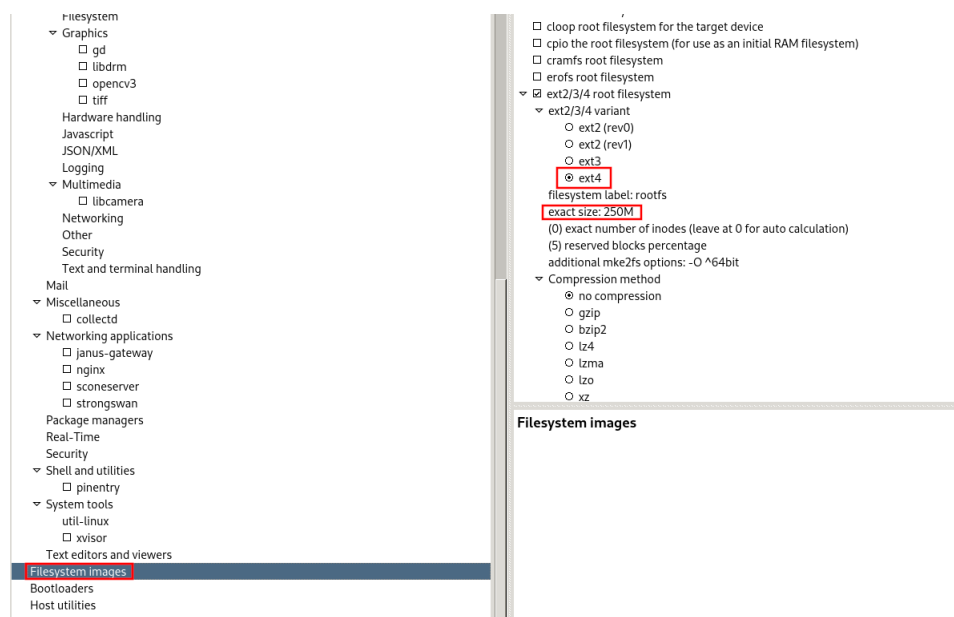


Figura 3.9: Filesystem Images

Capítulo 4

Configurações Kernel

Realizadas as configurações do buildroot, é necessário proceder a alterações no kernel. Essencialmente temos de especificar os drivers que são utilizados pela webcam, bem como os da placa gráfica para permitir o uso da mesma.

Dentro do diretório `buildroot` execute o seguinte comando:

```
$ make linux-xconfig
```

Em "Device Drivers":

- Selecione "Multimedia support" e habilite "Cameras/video grabbers support", Figura 4.1
- Em "Multimedia support > Media USB Adapters", Figura 4.2.
 - Selecione "UVC input events device support"
 - Em "Media USB Adapters > GSPCA based webcams" selecione os seguintes drivers, Figura 4.3:
 - * OV51X/OVFX2/W996xCF;
 - * OV534 OV772x;
 - * OV534 OV965x.

Ainda dentro de "Device Drivers":

- Em "Graphic support > Frame buffer Devices" selecione os drivers da **NVIDIA** (características coerentes com o hardware utilizado), Figura 4.4.

Em **Device Drivers > USB support** habilite a utilização dos dispositivos **USB 2.0** e **USB 3.0**, Figura 4.5.

Clique em salvar e feche a janela para prosseguir. Depois execute o **make** para compilar tudo.

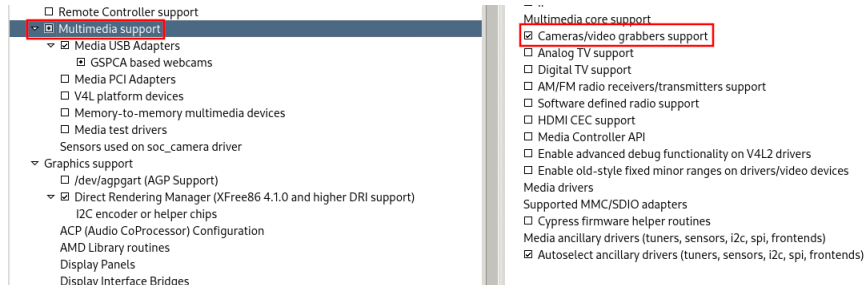


Figura 4.1: Multimedia Support

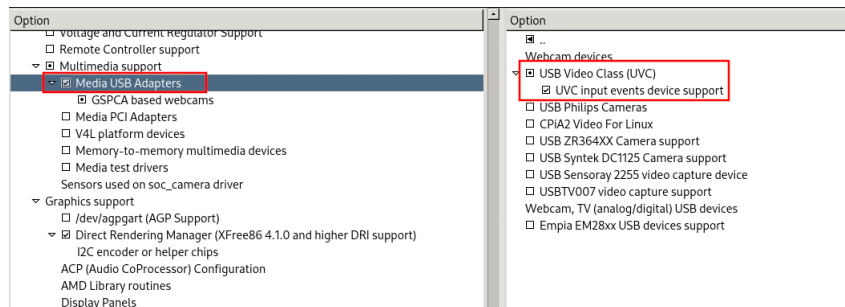


Figura 4.2: UVC

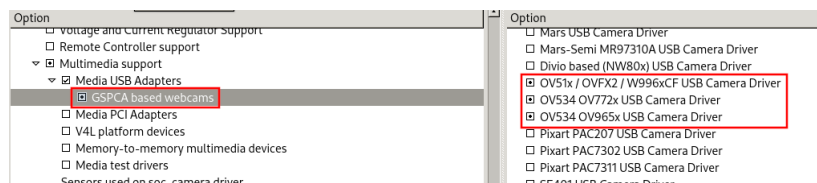


Figura 4.3: Câmara Drivers

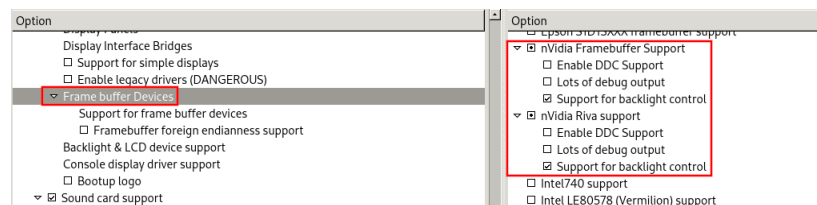


Figura 4.4: Graphic Drivers

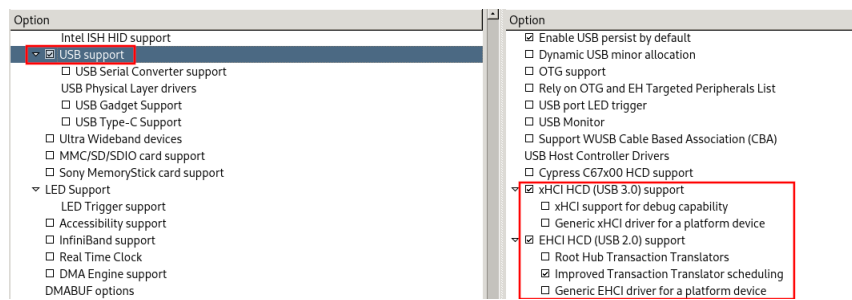


Figura 4.5: USB

Capítulo 5

QEMU

O QEMU permitiu realizar diversos testes importantes ao longo do projeto. Desde logo averiguar a compatibilidade dos drivers, bem como a conectividade com a internet.

```
$ qemu-system-x86_64 -kernel output/images/bzImage -append  
↪ "root=800" -enable-kvm -drive  
↪ file=output/images/rootfs.ext4,format=raw -usb -device  
↪ qemu-xhci,id=xhci -device  
↪ usb-host,bus=xhci.0,hostdevice=/dev/bus/usb/XXX/YYY
```

Através do comando `lsusb`, podemos verificar o bus **XXX** e o device **YYY** designados para o nosso dispositivo.

- **-kernel** aponta para o kernel compilado "bzimage"
- **file** aponta para a imagem do sistema "rootfs.ext4"

É importante notificar que durante a fase de ensaios, ocorreram erros que impediam a utilização da webcam quando esta não era propriamente desligada. De modo a evitar esse erro, utilizamos o seguinte conjunto de comandos [10]:

```
$ fuser /dev/video0  
$ kill -9 <ID_DO_PROCESSO>
```

Comandos

1. Iniciar a webcam no vlc [11]

```
$ cvlc v4l2:///dev/video0
```

2. Iniciar a gravação de um vídeo [12]:

```
$ ffmpeg -f video4linux2 -s 640x480 -i /dev/video0  
↪ -codec:v huffyuv -threads 1 <nome_vídeo>.mkv
```

- **-f** força o formato do ficheiro de input
- **-s** define o tamanho dos frames
- **-i** define o input (webcam)
- **-codec** define o codec a utilizar (huffyuv)
- **-threads** especifica o número de threads usadas pelo CPU

3. Retirar uma imagem de um vídeo em gravação [13]:

```
$ ffmpeg -f video4linux2 -s 640x480 -i /dev/video0  
↪ -codec:v huffyuv -threads 4 1.mkv -vf fps=0.5  
↪ <nome_imagem_output>.jpg
```

4. Retirar uma imagem de um vídeo previamente gravado:

```
$ ffmpeg -ss 0.5 -i <nome_vídeo>.mkv -t 1 -s 480x300  
↪ <nome_imagem_output>.jpg
```

5. Para visualizar os vídeos e imagens capturados utilizamos o seguinte comando:

```
$ cvlc <nome_ficheiro>
```

Capítulo 6

Dispositivo de armazenamento

Neste segmento, preparamos um dispositivo de armazenamento para que o mesmo passe a conter a imagem da distribuição criada.

Configuração

Dentro do diretório **buildroot** execute os seguintes comandos.

Identificação do dispositivo de armazenamento:

```
$ cat /proc/partitions
```

Verificar que todas as partições se encontram desmontadas:

```
$ su
$ umount /dev/sdb*
```

Repita até obter a seguinte mensagem:

```
$ umount: /dev/sdb: not mounted.
```

De seguida apagamos a tabela de partições existente, e criamos duas partições:

```
$ dd if=/dev/zero of=/dev/sdb bs=1M count=20
```

```
$ fdisk /dev/sdb
```

Para a primeira partição:

- Inserimos **'n'**, seguido de *Enter*
- Pressione *Enter* para criar a partição primária
- Pressione *Enter* para o criar como partição 1
- Para o primeiro setor, defina **8192**
- O último setor defina **+88471**
- Pressione **'t'** para mudar o tipo de partição. A primeira partição deve ser alterada para (W95 FAT32 (LAB)) pressionando **'c'**.
- Pressione **'a'** para partição de boot

Para a segunda partição:

- Inserimos **'n'**, seguido de *Enter*
- Pressione *Enter* para criar a partição primária
- Pressione *Enter* para o criar como partição 2
- Para o primeiro setor, defina **98304**
- O último setor, pressione *Enter* para usar todo o espaço restante

Pressione **'p'** para verificar a tabela de partições. E finalize com **'w'**, para aplicar as configurações.

A primeira partição deve conter um VFAT file system, para isso executa-se o seguinte comando:

```
$ wget
↪ http://ave.dee.isep.ipp.pt/~jes/arcom/Lab3-Buildroot/vfat.img.gz
$ gunzip vfat.img.gz
$ dd if=vfat.img of=/dev/sdb1
```

Abra o cmdline.txt para que o possa alterar, da seguinte forma:

```
$ mkdir m1
$ mount /dev/sdb1 m1
$ <editor_de_texto> m1/cmdline.txt
```

Copie para o cmdline.txt o texto seguinte:

```
dwc_otg.lpm_enable=0 console=serial0,115200
root=/dev/sda2 rootfstype=ext4 elevator=deadline fsck.repair=yes
↪ rootwait
```

Para formatar a segunda partição em EXT4, execute os seguintes passos:

```
$ mkfs.ext4 /dev/sdb2
$ exit
```

Agora voltando a primeira partição:

```
$ umount /dev/sdb*
$ syslinux -i /dev/sdb1
$ dd if=/usr/share/syslinux/mbr.bin of=/dev/sdb conv=fsync
$ umount /dev/sdb1
$ mkdir m1
$ mount /dev/sdb1 m1
$ cd m1
$ <editor_de_texto> syslinux.cfg
```

Sendo que neste ultimo comando escreva o seguinte para o ficheiro:

```
LABEL arcom
KERNEL bzImage
APPEND vga=0x315 root=802 rootdelay=5
```

De seguida execute novamente os seguintes comandos:

```
$ cp -a output/images/bzImage m1
$ umount m1
$ mkdir m
$ mount output/images/rootfs.ext4 m
$ mount /dev/sdb2 m1
$ cp -a m/* m1
$ umount m m1
$ umount /dev/sdb*
```

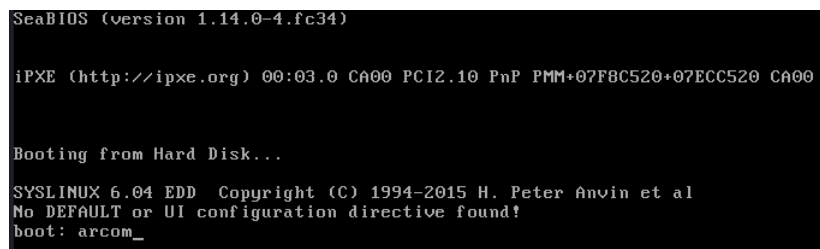
De forma a verificar esta secção, utilize o seguinte comando e realize novamente todos os passos indicados no capítulo 5:

```
$ qemu-system-x86_64 /dev/sdb -enable-kvm -usb -device
↪ qemu-xhci,id=xhci -device
↪ usb-host,bus=xhci.0,hostdevice=/dev/bus/usb/XXX/YYY
```


Capítulo 7

Resultados

O arranque pelo dispositivo de armazenamento foi o último teste realizado. Tendo-se utilizado o syslinux é agora necessário indicar no arranque da máquina o ficheiro que contem a imagem do sistema, neste caso indicando a nossa label, "**arcom**", Figura 7.1.



```
SeaBIOS (version 1.14.0-4.fc34)

iPXE (http://ipxe.org) 00:03.0 CA00 PCI2.10 PnP PMM+07F8C520+07ECC520 CA00

Booting from Hard Disk...

SYSLINUX 6.04 EDD Copyright (C) 1994-2015 H. Peter Anvin et al
No DEFAULT or UI configuration directive found!
boot: arcom_
```

Figura 7.1: Arranque do sistema

Para iniciar o streaming para o ip de uma segunda máquina dentro da mesma rede LAN. Executamos o seguinte comando:

```
$ ffmpeg -f v4l2 -i /dev/video0 -r 10 -b:v 2000k -s 640x480 -c:v  
↪ mpeg2video -f mpgets -flush_packets 1 udp://<ip_máquina_2>:5000
```

Iniciando o VLC na segunda máquina, seleccionamos a opção **Open Network Stream**, e inserimos o URL seguinte: `udp://@0.0.0.0:5000`, Figura 7.2 .

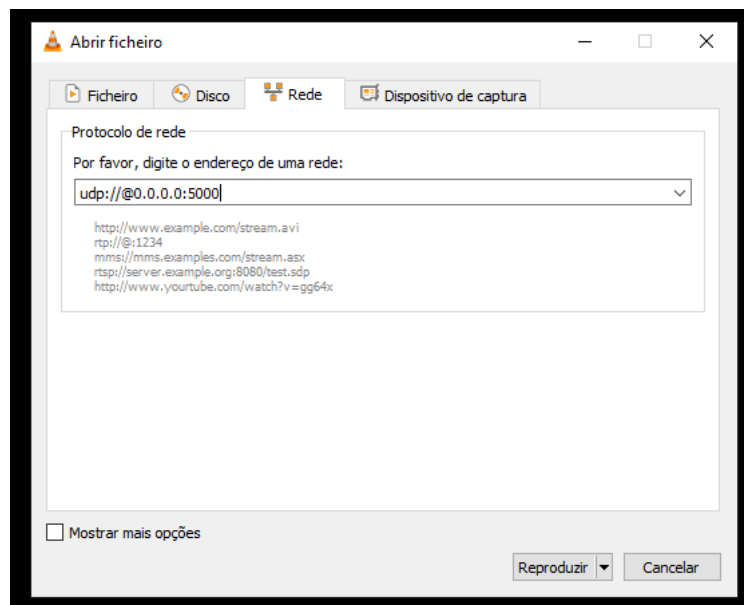


Figura 7.2: VLC Network

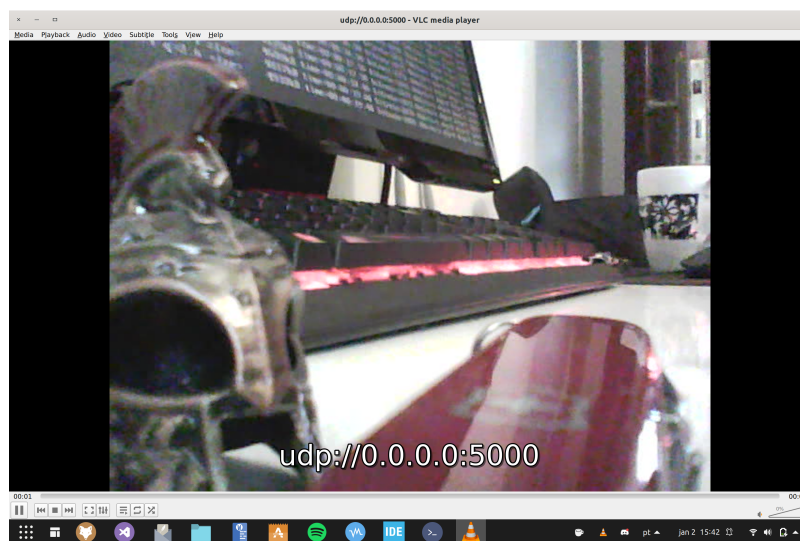


Figura 7.3: VLC com stream online

A secção seguinte inclui os dados relativos ao sistema: CPU (Figura 7.4), RAM (Figura 7.5), memória usada (**df**) (Figura 7.6) e memória usada (**du**) (Figura 7.7).

```
$ cat /proc/cpu info
$ free -m
$ df
$ du -k -d 1 -a
```

```

processor      : 11
vendor_id     : AuthenticAMD
cpu family    : 23
model         : 1
model name    : AMD Ryzen 5 1600 Six-Core Processor
stepping      : 1
microcode     : 0x8001138
cpu MHz       : 2722.927
cache size    : 512 KB
physical id   : 0
siblings      : 12
core id       : 6
cpu cores     : 6
apicid        : 13
initial apicid : 13
fpu           : yes
fpu_exception : yes
cpuid level   : 13
wp            : yes
flags         : fpu_ume de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mm
x fxsr sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl nonstop_t
sc cpuid extd_apicid aperfmperf pni pclmulqdq monitor sse3 fma cx16 sse4_1 sse4_2 xsave popcnt aes
xsaveaux fl6c rdrand lahf_lm cmp_legacy svm extapic cr0_legacy sha sse4a misalignsse 3dnowprefetch
osvw skinit wdt tce topoext perfctr_core perfctr_nb bpext perfctr_llc mwaitx cpb hu_pstate sme ssbd
seu ibpb ummcall fsgsbase bti1 aux2 smep bmi2 rdseed adx smap clflushopt sha_ni xsaveopt xsavec xget
bv1 xsavec clzero irperf xsaveerptr arat npt lbrv svm_lock nrip_save tsc_scale vmcb_clean flushbyas
d decodeassists pausefilter pfthreshold avic v_umsave_umsload ugif overflow_recov succor srca
bugs          : sysret_ss_attrs null_seg spectre_v1 spectre_v2 spec_store_bypass
bogomips      : 6399.17
TLB size      : 2560 4K pages
clflush size   : 64
cache_alignmen : 64
address sizes  : 43 bits physical, 48 bits virtual
power managem : ts ttp tm hupstate eff_freq_ro [13] [14]

```

Figura 7.4: Informação do CPU

```

# free -m

```

	total	used	free	shared	buff/cache	available
Mem:	112	16	60	0	35	89
Swap:	0	0	0			

Figura 7.5: RAM utilizada

```

# df

```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/root	7586440	189824	6991524	3%	/
devtmpfs	55896	0	55896	0%	/dev
tmpfs	57424	0	57424	0%	/dev/shm
tmpfs	57424	408	57016	1%	/tmp
tmpfs	57424	196	57228	0%	/run

Figura 7.6: Disco usada df

```
# du -k -d 1 -a
4      ./mnt
66796  ./lib
276    ./sbin
0      ./linuxrc
25900  ./root
48076  ./usr
8328   ./etc
196    ./run
0      ./sys
4      ./media
1388   ./bin
16     ./lost+found
368    ./tmp
84     ./var
0      ./proc
4776   ./boot
0      ./dev
4      ./opt
0      ./lib64
156220 .
#
```

Figura 7.7: Disco usada du

Capítulo 8

Conclusões

Tendo em conta o objetivo proposto deste projeto, o sistema de vigilância é então capaz de utilizar a webcam designada para efeitos de filmagem. Adicionalmente foram incluídas as funcionalidades de captura de imagem, quer em tempo real quer em ficheiros previamente gravados, e a possibilidade de streaming para máquinas dentro da mesma rede LAN.

Os principais contratempos com que nos deparamos prendem-se essencialmente com a versão do kernel linux utilizado, uma vez que para versões acima da 5.0 não foi possível habilitar os drivers da NVIDIA para serem usados na máquina de teste.

Foi ainda realizado um último teste, nos computadores das sala de laboratório, em que após escolher a opção de driver de gráfica e a sua resolução, no boot, foi possível obter os mesmos resultados demonstrados nos capítulos 6 e 7. Desta forma conseguimos validar que o sistema desenvolvido pode ser utilizado em diversos sistemas, desde que possuam características semelhantes, nomeadamente um processador com arquitetura x86_64, bem como uma componente gráfica para visualizar o output obtido pela câmara, portas USB e conectividade à Internet.

Referências

- [1] Armadeus, “Gspcawebcam.” <http://www.armadeus.org/wiki/index.php?title=GspcaWebcam>, 2013. Accessed: 2022-01-07. [Citado na página 1]
- [2] The kernel development community, “6.12. the gspca cards list.” <https://www.kernel.org/doc/html/v4.18/media/v4l-drivers/gspca-cardlist.html>, 2019. Accessed: 2021-12-10. [Citado na página 1]
- [3] Kalastros, “Camera driver support under linux.” <https://blog.katastros.com/a?ID=00300-27cc41ee-99f8-4de0-8a29-a8d3ded42b86>, 2020. Accessed: 2022-01-10. [Citado na página 2]
- [4] Wikipedia, “Vlc wiki.” <https://wiki.videolan.org/Documentation:Documentation>, 2021. Accessed: 2022-01-07. [Citado na página 2]
- [5] Wikipedia, “Huffyuv.” <https://pt.wikipedia.org/wiki/HuffyUV>, 2020. Accessed: 2022-01-09. [Citado na página 2]
- [6] ffmpeg developers, “ffmpeg.” <https://ffmpeg.org/ffmpeg.html#Description>, 2021. Accessed: 2022-01-07. [Citado na página 2]
- [7] buildroot developers, “buildroot.” <https://buildroot.org/download.html>, 2021. Accessed: 2022-01-07. [Citado na página 3]
- [8] Jorge Estrela, “Cross-compilation with buildroot.” http://ave.dee.isep.ipp.pt/~jes/arcom/Lab3-Buildroot/Lab3-LinuxDistro-Buildroot_and_disk_setup-en.pdf, 2021. Accessed: 2022-01-07. [Citado na página 5]
- [9] Jorge Estrela, “Linux distribution: Kernel configuration.” <http://ave.dee.isep.ipp.pt/~jes/arcom/Lab5-KernelCompilation/Lab5-Kernel-en.pdf>, 2021. Accessed: 2022-01-07. [Citado na página 5]
- [10] Aaron Kili, “Learn how to use ‘fuser’ command with examples in linux.” <https://www.tecmint.com/learn-how-to-use-fuser-command-with-examples-in-linux/>, 2021. Accessed: 2022-01-07. [Citado na página 15]
- [11] VLC Wiki, “Documentation:modules/v4l2.” <https://wiki.videolan.org/Documentation:Modules/v4l2/>, 2021. Accessed: 2022-01-07. [Citado na página 16]

- [12] ffmpeg Wiki, “Capture/webcam.” <https://trac.ffmpeg.org/wiki/Capture/Webcam>, 2021. Accessed: 2022-01-07. [Citado na página 16]
- [13] ffmpeg Wiki, “Image.” <https://trac.ffmpeg.org/wiki/Creathumbnailimageeverysecondsofthevideo>, 2021. Accessed: 2022-01-07. [Citado na página 16]