

Laboratório de Sistemas Operacionais

Trabalho Prático 1

*Fernando Varrone Giacomini
João Benno Weber
Igor Vicente*

1. Introdução:

Esse trabalho tem como objetivo personalizar uma distribuição de linux para executar um servidor Web contendo algumas informações do sistema.

2. Configuração da distro/buildroot

De começo, é preciso criar um diretório de trabalho, no nosso caso, o linuxdistro, onde a maioria dos passos serão executados.

em negrito, os comandos de linha a serem executados

mkdir linuxdistro
cd linuxdistro

° Faça download da versão 2022.02 do Buildroot através do site ou do comando de linha:
<https://buildroot.org/downloads/buildroot-2022.02.tar.gz>

wget --no-check-certificate https://buildroot.org/downloads/buildroot-2022.02.tar.gz

Descompacte o arquivo baixado para o diretório linuxdistro.

tar -zxvf buildroot-2022.02.tar.gz

Renomeie o diretório criado para buildroot (para simplificação)

mv buildroot-2022.02/ buildroot/

Vá para o diretório buildroot/ e execute o comando abaixo:

make qemu_x86_defconfig

Este comando aplicará as configurações padrões à distribuição, mas iremos customizar algumas delas.

Para isso, utilize o seguinte comando para entrar na interface de configuração do Buildroot:

make menuconfig

Agora, listamos as opções que precisam ser alteradas para o funcionamento da aplicação:

- **System configuration --->**

() Network interface to configure through DHCP

[*] Run a getty (login prompt) after boot --->

(ttyS0) TTY port

Apague a opção existente ao apertar "espaço" na opção (Run a getty...)
apague o conteúdo existente e escreva no lugar "ttyS0"

Habilite o suporte à linguagem C++ (isso será necessário para a execução do webserver):

Toolchain --->

[*] Enable C++ support

Saiu do menu de configurações salvando as opções.

Ainda no mesmo diretório, é preciso alterar as seguintes configurações do linux usando o comando:

make linux-menuconfig:

-Habilite o driver Ethernet1000:

Device Drivers --->

[*] Network device support --->

[*] Ethernet driver support --->

<*> Intel(R) PRO/1000 Gigabit Ethernet support

3. Configuração da interface de rede na dist

Para que possamos executar o programa http na máquina target e abrir o html desenvolvido através do host, iremos configurar o Buildroot para gerar a distribuição com as configurações de rede desejada.

O Buildroot permite a execução de scripts customizados pelo usuários em diferentes momentos da compilação. Neste caso, desejamos executar um script (*pre-build.sh*) para copiar o script de configuração da rede (*S41network-config*) antes da geração do sistema de arquivos (rootfs).

No diretório *custom-scripts* (criado anteriormente), crie um arquivo denominado *S41network-config* com o seguinte conteúdo:

```
#!/bin/sh
#
# Configuring host communication.
#

case "$1" in
start)
    printf "Configuring host communication."

    /sbin/ifconfig eth0 192.168.1.10 up
    /sbin/route add -host <IP-DO-HOST> dev eth0
    /sbin/route add default gw <IP-DO-HOST>
    [ $? = 0 ] && echo "OK" || echo "FAIL"
    ;;
stop)
    printf "Shutdown host communication. "
    /sbin/route del default
```

```

        /sbin/ifdown -a
        [ $? = 0 ] && echo "OK" || echo "FAIL"
        ;;
restart|reload)
    "$0" stop
    "$0" start
    ;;
*)
    echo "Usage: $0 {start|stop|restart}"
    exit 1
esac

exit $?

```

*Lembrando de substituir <IP-DO-HOST> pelo IP real do host (é possível verificar o ip do host através do comando **sudo ifconfig**).

Ainda no diretório *custom-scripts*, crie um novo arquivo chamado *pre-build.sh*, com o código abaixo.

```

#!/bin/sh

cp$BASE_DIR/../custom-scripts/S41network-config $BASE_DIR/target/etc/init.d

chmod +x $BASE_DIR/target/etc/init.d/S41network-config

```

De permissão de execução para o script *pre-build.sh*.

```

chmod +x custom-scripts/pre-build.sh

```

Agora, é preciso configurar o buildroot para que o script *pre-build* seja executado antes da imagem do rootfs

```

$ make menuconfig

```

System configuration --->

(custom-scripts/pre-build.sh) Custom scripts to run before creating filesystem images

Feche a interface, salvando as alterações feitas.

4. Servidor Web

Antes do desenvolvimento da aplicação, é preciso permitir que o compilador cruzado seja disponibilizado na variável de ambiente PATH:

```

$ export PATH=$PATH:~/linuxdistro/buildroot/output/host/bin

```

Para fazer o servidor usamos como base o código em C disponibilizado no enunciado. As informações do sistema são coletadas usando a função **popen()** para executar comandos no shell do linux, como apresentado a seguir na Figura 1:

```
/*----TIME----*/
strcat(page,p);
    fp = popen("date", "r");
    while((a=fgetc(fp))!= EOF){
        //printf("%c",a); //PRINT
        cToStr[0] = a;
        cToStr[1] = '\0';
        strcat(page,cToStr);
    }
    pclose(fp);
strcat(page,p_end);
```

Figura 1: Exemplo do trecho de código que coleta o tempo

As informações são concatenadas na string page junto com o resto do código HTML. As informações são atualizadas toda vez que a página é recarregada.

O código completo do web server após as alterações está disponível em: <https://github.com/iMarVic/Web-Server-T1-LabSisOp>.

O compilador cruzado é necessário para a execução do arquivo em binário da aplicação, junto de bibliotecas adicionais, na máquina target seja realizado. Através do seguinte comando:

\$ i686-linux-gcc (nome arquivo aplicação) -O2 -o (nome arquivo binário aplicação)

O arquivo binário gerado deve então ser colocado na pasta da distribuição:

buildroot/output/target/usr/bin/

Por fim, basta executar a emulação do guest, através do comando:

```
$ sudo qemu-system-i386 --device e1000,netdev=eth0,mac=aa:bb:cc:dd:ee:ff \
    --netdev tap,id=eth0,script=custom-scripts/qemu-ifup \
    --kernel output/images/bzImage \
    --hda output/images/rootfs.ext2 \
    --nographic \
    --append "console=ttyS0 root=/dev/sda"
```

Agora, execute a aplicação na máquina target, digitando o nome do arquivo em binário adicionado na pasta *“buildroot/output/target/usr/bin/”*

No terminal da máquina target, a seguinte mensagem deve aparecer:

"Waiting a connection..."

Para realizar a visualização da página html, basta digitar em qualquer browser a seguinte url:

<IP DO TARGET>:8000

Com isso, agora é possível visualizar, em tempo real, as seguintes especificações da máquina target através da máquina host:

- Data e hora do sistema;
- Uptime (tempo de funcionamento sem reinicialização do sistema) em segundos;
- Modelo do processador e velocidade;
- Capacidade ocupada do processador (%);
- Quantidade de memória RAM total e usada (MB);
- Versão do sistema;
- Lista de processos em execução (pid e nome).

5. Conclusão

Ao decorrer do desenvolvimento do trabalho foi possível acompanhar o processo de criação e configuração de uma distribuição linux como: plataforma x86, configuração de rede, driver Ethernet e etc, seguindo o tutorial disponibilizado na área moodle da disciplina com a descrição dos passos necessários.

Após a criação da distribuição linux e a devida configuração da rede da mesma foi possível rodar um *web server* na distribuição criada, capaz de apresentar uma página HTML na máquina *host* com os dados solicitados no exercício, como horário do sistema e suas configurações, por exemplo.

Por fim temos o seguinte cenário: uma distribuição Linux rodando um *web server* que disponibiliza através da conexão pela interface de rede o acesso pela máquina *host* para acessar dados do sistema na página HTML gerada pelo servidor.