**I. Topic and expected final product of the project**

**1. Topic**

\* Research , learn and demo how to use BusyBox

**2. Expected final product:**

\* A webserver - building by httpd utility of BusyBox and cgi-script that :

- Provide an interface that showing the list of files and folders that serve by the httpd server

- Allow client download file and folder that server by the server

- Support basic authentication (username and password) , allow the owner to :

+ Upload file

+ Create folder

+ Rename file

+ Edit the text file

+ Delete files/folders

+ Copy files/folders to a directory

- Deny, allow access from specific IP addresses.

\* Deploy and test webserver on a Linux-based System (built by QEMU and Busybox)

**II. Introduction to Busybox , QEMU and httpd**

**1. BusyBox - The Swiss Army Knife of Embedded Linux**

**Ảnh có chứa côngtenơ, hộp, các tông, bìa các tông

Mô tả được tạo tự động**

Pic 1: Logo of BusyBox

**1.1. What is BusyBox**

- BusyBox is a software package that provides a collection of common UNIX utilities in a single executable file

- It is designed to be lightweight and efficient, making it well-suited for embedded systems and resource-constrained environments

**1.2 Why we need BusyBox:**

-In a regular Linux distribution, each command (such as ls,ps,…) corresponds to a separate program. Busybox combines all of these commands into a single program, residing in a single file named "busybox"​.

=>The commands cp, ls, cat, etc. are symbolic links to this single Busybox file

- Also BusyBox remove some rarely used or unnecessary command

=> **BusyBox help saving storage space in embedded systems which are typically limited in terms of storage capacity and memory.**



Pic 2 : Size of /bin in Ubuntu – about 200 MB Pic 3: Size of /bin (busybox) : 2.6 M

**2 . httpd and cgi-script**

**2.1. httpd**

- httpd (HTTP Daemon) is an utility of BusyBox .Function of httpd is listen for incoming HTTP

- httpd help us to build a lightweight HTTP webserver

- Enables us to interact with the embedded system through a web browser to perform administrative tasks, update settings

**2.2 cgi-script**

-CGI (Common Gateway Interface) script is an interface specification that enables web servers to execute an external program, typically to process user requests

Ảnh có chứa văn bản, thiết kế, ảnh chụp màn hình

Mô tả được tạo tự động

Pic 4 :How httpd server work with cgi-script

(Source: <https://www.loginsoft.com/>)

-Why we need CGI (bash) script? ​

+HTML only can show the static content​

+We want to build a web server to serve file and let client do some action on the file​

=> List of file must be generate dynamically,not statically ​

- Busybox httpd can execute CGI scripts ,allow us to run dynamic web application , ​even perform some action on file/folder

**3. QEMU**

****

Pic 5 : Logo of QEMU

-QEMU (Quick Emulator) is an open-source virtualization software that allows us to emulate and run operating systems and software on various hardware platforms

-It provides a framework for full system emulation, enabling us to create virtual machines and run different operating systems within them.

**III. Project Implementation:**

**1. Build a minimal Linux from scratch using QEMU and Busybox:**

\* To build a minimal Linux from scratch need 2 key artifact : Kernel Image and Disk Image

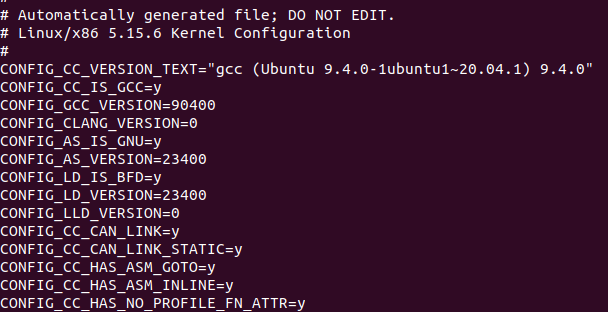
**- (Build) Kernel Image:**

+Download from source :

wget <https://mirrors.edge.kernel.org/pub/linux/kernel/v5.x/linux-5.15.6.tar.xz>

+Decompress : **tar -xf linux-5.15.6.tar.xz**

+Create default kernel configuration file : **make defconfig**



Pic 6: Some first row of .config file

Compile the kernel : make

Finally , we can have a kernel image : **bzImage** file in arch/x86/boot

Ảnh có chứa văn bản, phần mềm, Phần mềm đa phương tiện, ảnh chụp màn hình

Mô tả được tạo tự động

Pic 7: Location of bzImage after compile kernel successfully

**- Disk Image (init root device)**

**(+ Includes all program to be run**

**+ Used by kernel as root of Virtual File System)**

+ Download busybox from source

wget <https://busybox.net/downloads/busybox-1.34.1.tar.bz2>

+ Decompress : **tar -xf busybox-1.34.1.tar.bz2**

+ Create configuration file :**make menuconfig** ,choose build static binary

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Pic 8: Menu config of busybox

+ Compile busybox : make

+ After compile ,we will get the single executable binary file busybox



Pic 9: Size of busybox

+ Create initrd folder and its subfolder : bin proc sys dev home

+ Create symbolic link for various command to busybox in bin

Ảnh có chứa văn bản, Phông chữ, màu trắng

Mô tả được tạo tự động

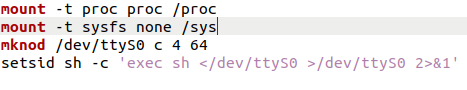
Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Pic 10-11: symbolic link to /bin/busbox

+Create CPIO archive with newc format (commonly used for Linux initial RAM disks) : **find . | cpio -o -H newc > ../initrd.img**

**-Create init file :**



Mount proc filesystem (interface to kernel data structures and system information) to /proc directory: **mount -t proc proc /proc**

Mount sysfs filesystem (system's device hierarchy) to the /sys directory :

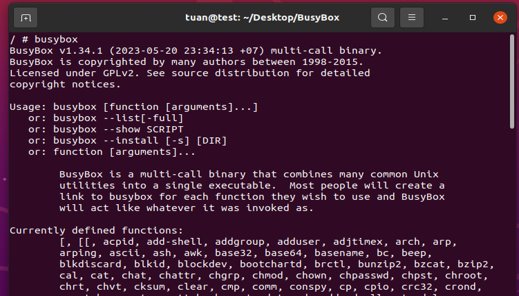
**mount -t sysfs none /sys**

Creates a character device node : **mknod /dev/ttyS0 c 4 64**

Start a new shell session : **setsid sh -c 'exec sh </dev/ttyS0 >/dev/ttyS0 2>&1'**

**- Build minimal Linux :**

**sudo qemu-system-x86\_64 -kernel bzImage -initrd initrd.img -nographic**

****

Pic 12: Result of building Linux

**2 .Build webserver**

**2.1 .Show the list of the file dynamically: (list\_files.sh)**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, tài liệu

Mô tả được tạo tự động

Pic 13: Content of list\_files.sh

Idea :

- Using for loop to get all the file and folder of current location :

**for file in "$path"/\*; do**

- Using some command get information of each file/folder:

+ **du -sh** : Get the file/folder size

+ **stat -c %U** : Get owner of the file/folder

+ **stat -u %y** : Get the latest date modify of the file

-If we click the folder , it will show all the file/folder of that directory by call the list\_files.sh again with new query path:

**if [ -d "$file" ]; then**

**echo "<td><a href=\"list\_files.sh?path=/$relativePath\">$filename</a></td>"**

**2.2 Download file/folder**

Idea

-For file : Using download attribute of html <a> tag

**<a href=\"$relativeLink\" download>Download</a>**

-For folder :**Write download.sh**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Pic 14: Step of download folders

+ Compress the folder first : tar -czf "download.tar.gz" "$folder\_path"

+ Send the content to client : cat "download.tar.gz"

+ Finally Remove the temporary file download.tar.gz



Pic 15: Content-type for downloading

+ The content-type must be application/octet-stream to indicate that this is a arbitrary binary file and send to client as an attachment using “Content-Disposition”

**2.3 Upload file :**

Idea : Using html form



Pic 16: html form for uploading

Client upload the file through <input type=file> and then submit to server using “post” method.

Server will read the content of the uploaded file then write it to a new file:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Pic 17: How to get file name

To get the filename of the uploaded file ,we can read each line of the content and find the “filename” and get the value

**2.4 Edit file :**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, đại số

Mô tả được tạo tự động

Pic 18: Text area to edit file

-First copy the current file content to a text area

-Then using html form to submit the new content and trigger the action handlesave.sh

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, hàng

Mô tả được tạo tự động

Pic 19: Handlesave script

handlesave.sh will read the content will parse the content of textarea (using **httpd -d** command) and overwrite to the file using :echo **"$CONTENT" > "$FILE\_PATH"**

**2.5 Other action: Simply use basic command :**

-Rename a file/folder (rename.sh): **mv $folder\_path $new\_path**

-Delete file/folder (remove.sh): **rm -rf $folder\_path**

-Copy (copy.sh) : **cp -r "$src" "$des"**

-Create folder : **mkdir -p**

-Delete multiple file/folder : Using checkbox (<input type="checkbox">) ,then using for loop to delete the chosen file/folder:

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình, đại số

Mô tả được tạo tự động

Pic 20: Function inside <script> tag

-The action rename,delete,create folder are written as function in <script> tag they will call to the corresponding cgi-script by using fetch() . Finally reload the page **location.reload()** to show the new change

**2.6 Simple authentication and Deny IP Address Access**

**Define in httpd.conf**

**Ảnh có chứa văn bản, ảnh chụp màn hình, tài liệu, Phông chữ

Mô tả được tạo tự động**

Pic 21: Template of file httpd.conf

**\* Syntax for authentication : path:<username>:<password>**

Eg : /cgi-bin/delete.sh:admin:123456

**\* Syntax for deny IP access: D: <IP address>**

Eg: D:10.0.2.16

**3.Deployment:**

**3.1 Set up bridge network**

-Creates a new bridge interface br0.

**sudo ip link add br0 type bridge**

- Adds a new TAP (network tap) interface tap0

**sudo ip tuntap add dev tap0 mode tap**

-Adds the enp0s3 interface to the bridge interface br0, making it a member of the bridge

**sudo ip link set dev enp0s3 master br0**

-Adds the tap0 interface to the bridge interface br0, making it another member of the bridge

**sudo ip link set dev tap0 master br0**

-Brings up the bridge interface br0, activating it and enabling traffic flow through it**.**

**sudo ip link set br0 up**

- Removes the IP address 10.0.2.15 from the enp0s3 interface

**sudo ip address delete 10.0.2.15 dev enp0s3**

-Assigns the IP address 10.0.2.15 with a subnet mask of /24 to the bridge interface br0.

**sudo ip address add 10.0.2.15 /24 dev br0**

-Adds a default route to the IP routing table, directing all traffic to the default gateway with the IP address 10.0.2.2, through the bridge interface br0

**sudo ip route add default via 10.0.2.2 dev br0**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Pic 22: Network interface after add

**3.2 Append the init file for starting webserver :**



Pic 23: New content of init file

-Assigns the IP address 10.0.2.55 to the eth0 interface and bring it up :

**ifconfig eth0 10.0.2.55 up**

-Assigns loopback IP address 127.0.0.1 to the lo (loopback) interface andbring it up :

**ifconfig lo 127.0.0.1 up**

-Start the webserver at port 8080 host all file in BusyBoxWebServer with the configuration define at /home/Config/httpd.conf

**httpd -p 8080 -h /home/BusyBoxWebServer/ -c /home/Config/httpd.conf**

**3.3 Rebuild the Linux-kernel based VM:**

**sudo qemu-system-x86\_64 -kernel bzImage -initrd initrd.img -nographic -net nic -net tap,ifname=tap0**

**-net nic:** creates a virtual network card that can be connected to a network.

**-net tap,ifname=tap0 :**connects the virtual NIC to tap0

**IV. Result and Test**

\* Open FireFox, enter 10.0.2.55:8080/cgi-bin/list\_files.sh?path=/.We can see the result below

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Pic 23: Webserver interface

\* Test Authentication: Enter button delete ,there will be a prompt

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Pic 24: Authentication prompt

-That means authentication works and we have to fill correctly to perform that action

**\* Test Deny IP Address Access:**

-Change the IP to 10.0.2.16:

+ sudo ip address delete 10.0.2.15 dev br0

+ sudo ip address add 10.0.2.16/24 dev br0

- Result :

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Trang web

Mô tả được tạo tự động

The page show 403 Forbidden ,it mean the deny IP address function work well

**V. Task Assignment**

|  |  |  |
| --- | --- | --- |
| Name | StudentId | Task |
| Nguyễn Minh Tuấn | 20194876 | -Compile kernel (bzImage) and busybox  -Write script to  + Download file/folder  + Modify text file  + Upload file  + Delete file/folder ,multiple file/folder |
| Nguyễn Thụ Hiếu | 20194761 | -Write init file to mount file system ,networking and start httpd server  -Write script to  + List file  + Search file  + Rename file  + Create Folder |
| Lê Đức Huy | 20194777 | - Set up bridge network and test authentication , IP access/deny  - Write  + Script to copy file ,select folder to copy  + httpd.conf file to support authentication |