#include<stdio.h>

#include<sys/time.h>

#include<sys/types.h>

#include<unistd.h>

#include<sys/stat.h>

#include<fcntl.h>

#include <linux/input.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <stdlib.h>

#define \_GNU\_SOURCE /\* See feature\_test\_macros(7) \*/

#include <stdio.h>

#define NAME "TOUCH\_SCREEN\_REPORT"

#define TYPE "EV\_SYB"

int main()

{

int server\_sockfd, client\_sockfd;

int server\_len, client\_len,ret;

struct sockaddr\_in server\_address;

struct sockaddr\_in client\_address;

server\_sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

server\_address.sin\_family = AF\_INET;

server\_address.sin\_addr.s\_addr = inet\_addr("192.168.1.37");

server\_address.sin\_port = htons(80);

server\_len = sizeof(server\_address);

ret=bind(server\_sockfd, (struct sockaddr \*)&server\_address, server\_len);

printf("ret value %d",ret);

perror("bind");

listen(server\_sockfd, 5);

char buffer\_1 [4095];

int i,rd,fd1,retfd;

char buf[4096];

int val;

char name[256] = "Unknown";

struct input\_event ev[64];

fd1=open("/dev/input/event0",O\_RDONLY);

if(fd1<0)

printf("error while open the file event1\n");

ioctl(fd1, EVIOCGNAME(sizeof(name)), name);

printf("Input device name: \"%s\"\n", name);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*server waiting\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

client\_len = sizeof(client\_address);

client\_sockfd = accept(server\_sockfd,

(struct sockaddr \*)&client\_address, &client\_len);

write(client\_sockfd,"<!DOCTYPE html>\n",15 );

write(client\_sockfd,"<html>\n",6 );

write(client\_sockfd,"<body>\n",6);

// write(client\_sockfd,"<style>",7);

// write(client\_sockfd,"body{background-color:lightblue;}\n",33);

// write(client\_sockfd,"h1{color:black;text-align:center;}\n",34);

// write(client\_sockfd,"</style>",7);

// write(client\_sockfd,"</head>",7);

// write(client\_sockfd,"<body>",6);

// write(client\_sockfd,"<h1>IoT\_HMI</h1>",7);

// write(client\_sockfd,"<p1>",4);

while(1)

{

rd=read(fd1, ev, sizeof(struct input\_event) \* 64);

if (rd < (int) sizeof(struct input\_event)) {

printf("yyy\n");

perror("\nevtest: error reading");

return 1;

}

for (i = 0; i < rd / sizeof(struct input\_event); i++)

{

if(ev[i].type == EV\_SYN)

{

sprintf(buffer\_1,"Event: time %ld.%06ld--------------%s------------\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec,NAME);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

else

{

if(ev[i].code==47){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_ABS), code %d (ABS\_MT\_SLOT), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

if(ev[i].code==53){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_ABS), code %d (ABS\_MT\_POSITION\_X), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

if(ev[i].code==54){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_ABS), code %d (ABS\_MT\_POSITION\_Y), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

if(ev[i].code==57){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_ABS), code %d (ABS\_MT\_TRACKING\_ID), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

if(ev[i].code==0){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_ABS), code %d (ABS\_X), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

if(ev[i].code==1){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_ABS), code %d (ABS\_Y), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

if(ev[i].type==1){

sprintf(buffer\_1,"Event: time %ld.%06ld, type %d (EV\_KEY), code %d (BTN\_TOUCH), value %d\n",ev[i].time.tv\_sec, ev[i].time.tv\_usec, ev[i].type,ev[i].code, ev[i].value);

write(client\_sockfd,buffer\_1,1024);

write(client\_sockfd,"\n\n",2);

}

}

//write(client\_sockfd,"<!DOCTYPE html>",15 );

//write(client\_sockfd,"<html>",6 );

//write(client\_sockfd,"<body>",6);

//close(client\_sockfd);

}

}

}