БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра технологий программирования

**Гришкин Андрей Иванович**

Отчёт по лабораторной работе №4

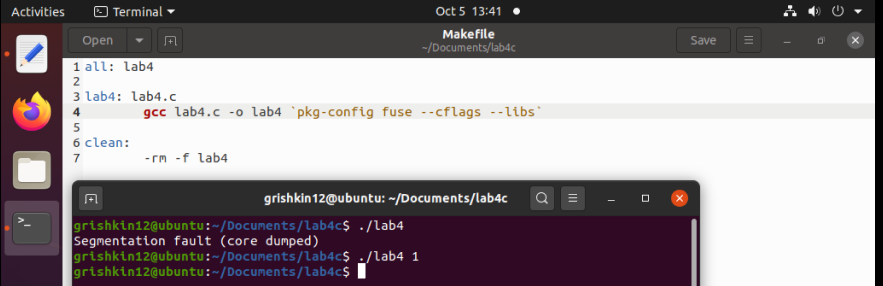
По курсу “Программирование мобильных и встраиваемых систем”

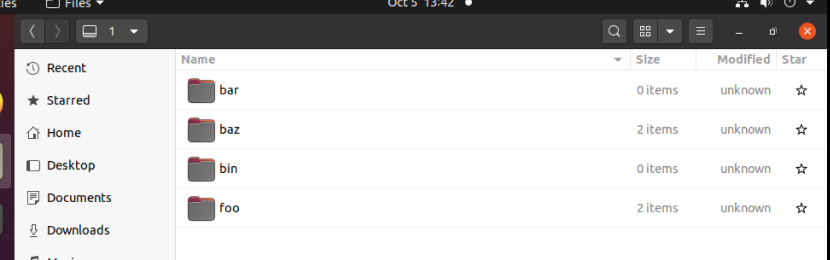
Базовые команды для работы с ОС LINUX для мобильных и

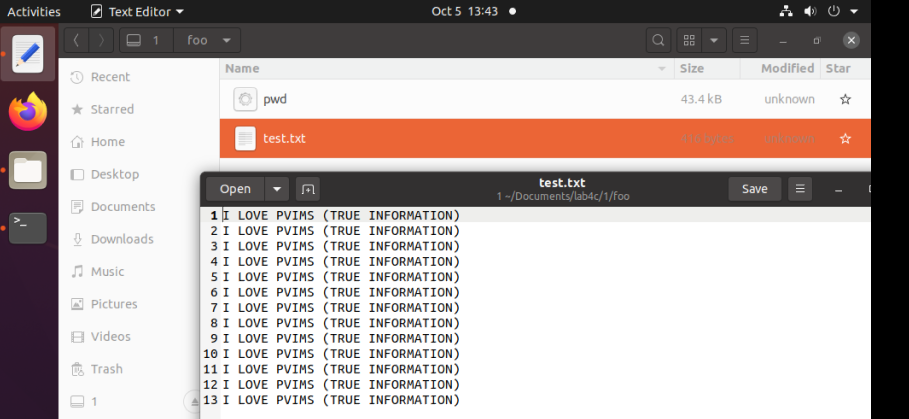
встраиваемых платформ

Преподаватель: Давидовская М.И.

Минск, 2020 г.

Пример makefile и работы программы 





Код

#define FUSE\_USE\_VERSION 30

#define \_FILE\_OFFSET\_BITS 64

#include <fuse.h>

#include <unistd.h>

#include <sys/types.h>

#include <stdio.h>

#include <string.h>

#include <errno.h>

#include <fcntl.h>

const int N=13;

char file\_system\_path[256] = "";

char\* example\_txt = "Hello world! Student Grishkin Andrei,1823113";

char test\_txt[13][50];

char\* readme\_txt = "Student Grishkin Andrei,1823113";

static int get\_attribute(const char\* path, struct stat \*st){

st->st\_uid = getuid();

memset(st,0,sizeof(struct stat));

if(strcmp(path,"/") == 0){

st->st\_mode = S\_IFDIR | 0755;

}else if(strcmp(path,"/bin") == 0){

st->st\_mode = S\_IFDIR | 022;

}else if(strcmp(path,"/bar") == 0){

st->st\_mode = S\_IFDIR | 555;

}else if(strcmp(path, "/baz") == 0){

st->st\_mode = S\_IFDIR | 744;

}else if(strcmp(path, "/baz/readme.txt") == 0){

st->st\_mode = S\_IFREG | 644;

st->st\_size = strlen(readme\_txt);

}else if(strcmp(path, "/baz/example") == 0){

st->st\_mode = S\_IFREG | 222;

st->st\_size = strlen(example\_txt);

}else if(strcmp(path,"/foo") == 0){

st->st\_mode = S\_IFDIR | 711;

}else if(strcmp(path,"/foo/pwd") == 0){

st->st\_mode = S\_IFREG | 777;

struct stat buffer;

stat("/usr/bin/pwd",&buffer);

st->st\_size = buffer.st\_size;

}else if(strcmp(path,"/foo/test.txt") == 0){

st->st\_mode = S\_IFREG | 000;

int size = 0;

for(int i = 0;i<N;i++){

size+=strlen(test\_txt[i])+1;

}

st->st\_size = size;

}else{

return -ENOENT;

}

return 0;

}

static int read\_directory(const char\* path, void\* buf, fuse\_fill\_dir\_t filler, off\_t offset, struct fuse\_file\_info \* fi){

filler(buf,".",NULL,0);

filler(buf,"..",NULL,0);

if(strcmp(path,"/") == 0){

filler(buf,"bin",NULL,0);

filler(buf,"bar",NULL,0);

filler(buf,"baz",NULL,0);

filler(buf,"foo",NULL,0);

if(strcmp(file\_system\_path,"/mnt/fuse/")!=0){

filler(buf,file\_system\_path+1,NULL,0);

}

return 0;

}else if(strcmp(path,"/bin") == 0){

return 0;

}else if(strcmp(path,"/bar") == 0){

return 0;

}else if(strcmp(path,"/baz") == 0){

filler(buf,"example",NULL,0);

filler(buf,"readme.txt",NULL,0);

return 0;

}else if (strcmp(path,"/foo") == 0){

filler(buf,"pwd",NULL,0);

filler(buf,"test.txt",NULL,0);

return 0;

}

return -ENOENT;

}

static int read\_file(const char\* path, char\* buf, size\_t size,off\_t offset, struct fuse\_file\_info\* fi){

size\_t length;

char\* read\_data;

if(strcmp(path,"/baz/readme.txt") == 0){

length = strlen(readme\_txt);

read\_data = readme\_txt;

}else if(strcmp(path,"/baz/example") == 0){

length = strlen(example\_txt);

read\_data = example\_txt;

}else if(strcmp(path,"/foo/pwd") == 0){

struct stat touch\_stat;

stat("/usr/bin/pwd",&touch\_stat);

length = touch\_stat.st\_size;

FILE\* f = fopen("/usr/bin/pwd","rb");

unsigned char buffer[length];

fread(buffer,length,1,f);

read\_data = buffer;

fclose(f);

}else if(strcmp(path,"/foo/test.txt") == 0){

char temp[N\*50+400];

strcpy(temp,"");

for(int i = 0;i<N;i++){

strcat(temp,test\_txt[i]);

strcat(temp,"\n");

}

length = strlen(temp);

read\_data = temp;

}else{

return -ENOENT;

}

if(offset<length){

if(offset+size>length){

size = length-offset;

}

memcpy(buf, read\_data+offset,size);

return size;

}

return 0;

}

int main(int argc, char\*\* argv){

strcpy(file\_system\_path,argv[1]);

char number\_line[8];

for(int i = 0;i<N;i++){

sprintf(number\_line,"%d: ",i+1);

strcat(test\_txt[i],"I LOVE PVIMS (TRUE INFORMATION)");

}

struct fuse\_operations operations = {

.getattr = get\_attribute,

.readdir = read\_directory,

.read = read\_file,

};

return fuse\_main(argc,argv,&operations,NULL);

}