УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

**“ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО”**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 11

по дисциплине «Операционные системы»

На тему **«**Управление файловой системой»

Выполнил: студент гр. ИТП-11

Клевцов А.О.

Принял: преподаватель-стажер

Карась О. В.

Гомель 2022

**Цель**: реализовать вспомогательную утилиту для файловой системы.

**Ход работы**

**Задание 1.** Реализовать вспомогательную утилиту для управления файловой системой.

В соответствии с вариантом необходимо реализовать утилиту «Управление доступом». Для задания была взята файловая система из 9 лабораторной работы и дополнена возможностями управления доступом: можно прочитать и изменить значения блока памяти.

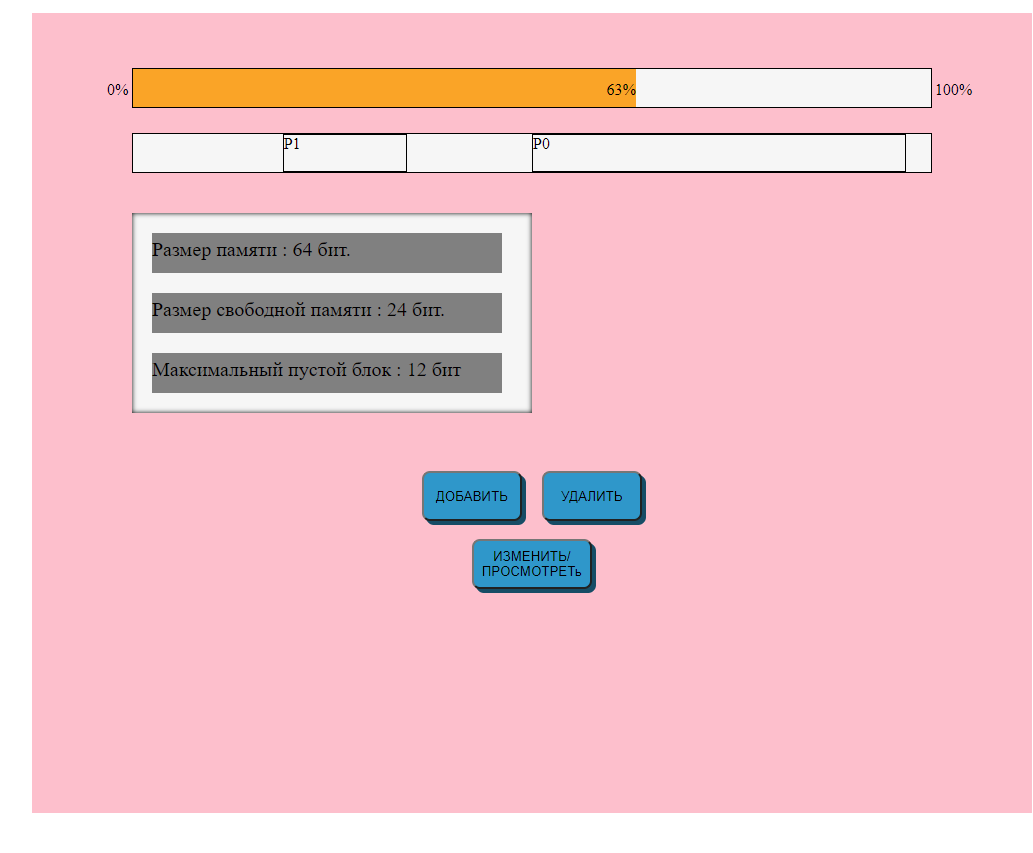


Рисунок 1 – Начальное окно

При запуске программы будет запрошено количество выделяемой памяти. После введения появится начальное окно, которое представлено на рисунке 1, в котором будет описана информация о памяти и блоках в этой памяти. Так же там присутствуют кнопки добавления и удаления блоков, всё это из предыдущей лабораторной работы номер 9. Новыми функциями является управление доступом к блокам.

По нажатию на кнопку изменить/ просмотреть будет запрошен номер блока, который требуется просмотреть, после этого появится прямоугольная область, в которой будет представлена информация про выбранный блок, а также будет возможность тут же изменить значение этого блока, введя текст в специальное поле, после нажатия на кнопку сохранить значение блока изменится. Так же можно просто скрыть окно и кнопку сохранить, нажав на крестик в правом верхнем углу.

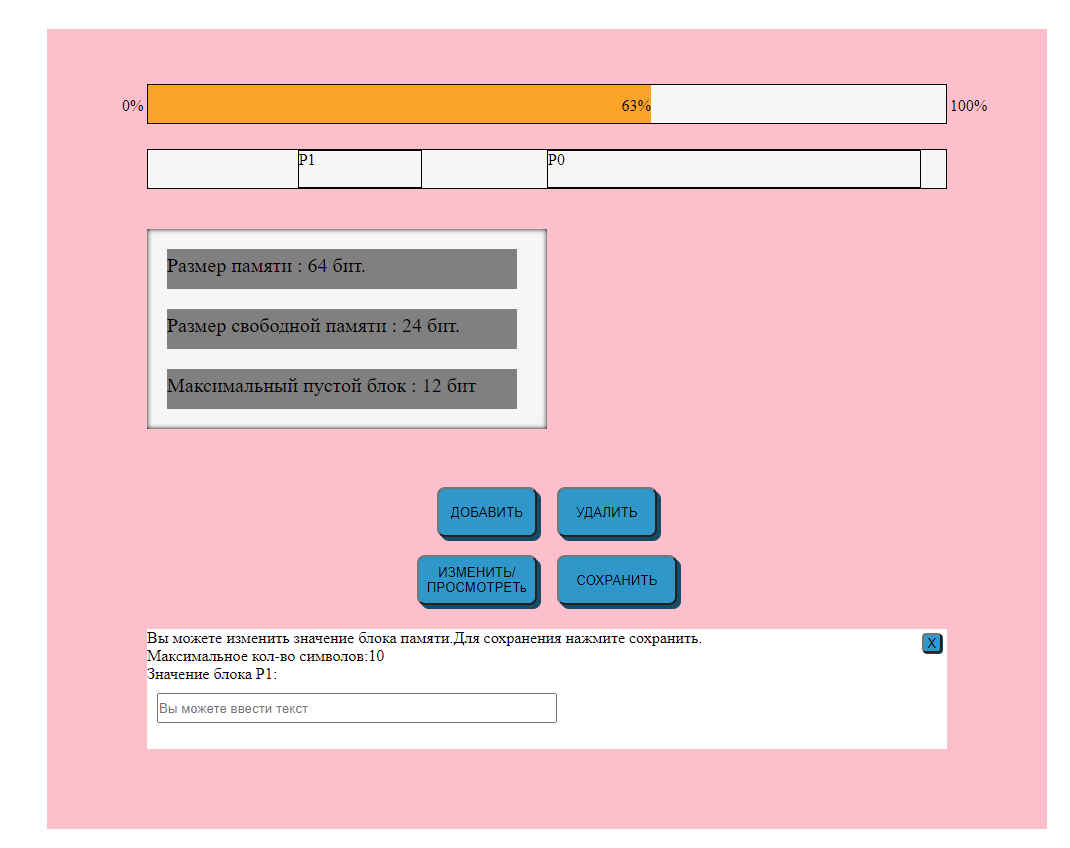


Рисунок 2 – Просмотр значения 1 блока

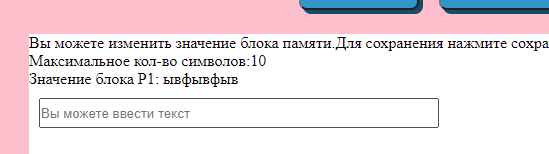


Рисунок 3 – Просмотр значения 1 блока после изменения

В случае если вводимый текст в поле будет превышать размер блока, то будет выведено сообщение об ошибке.

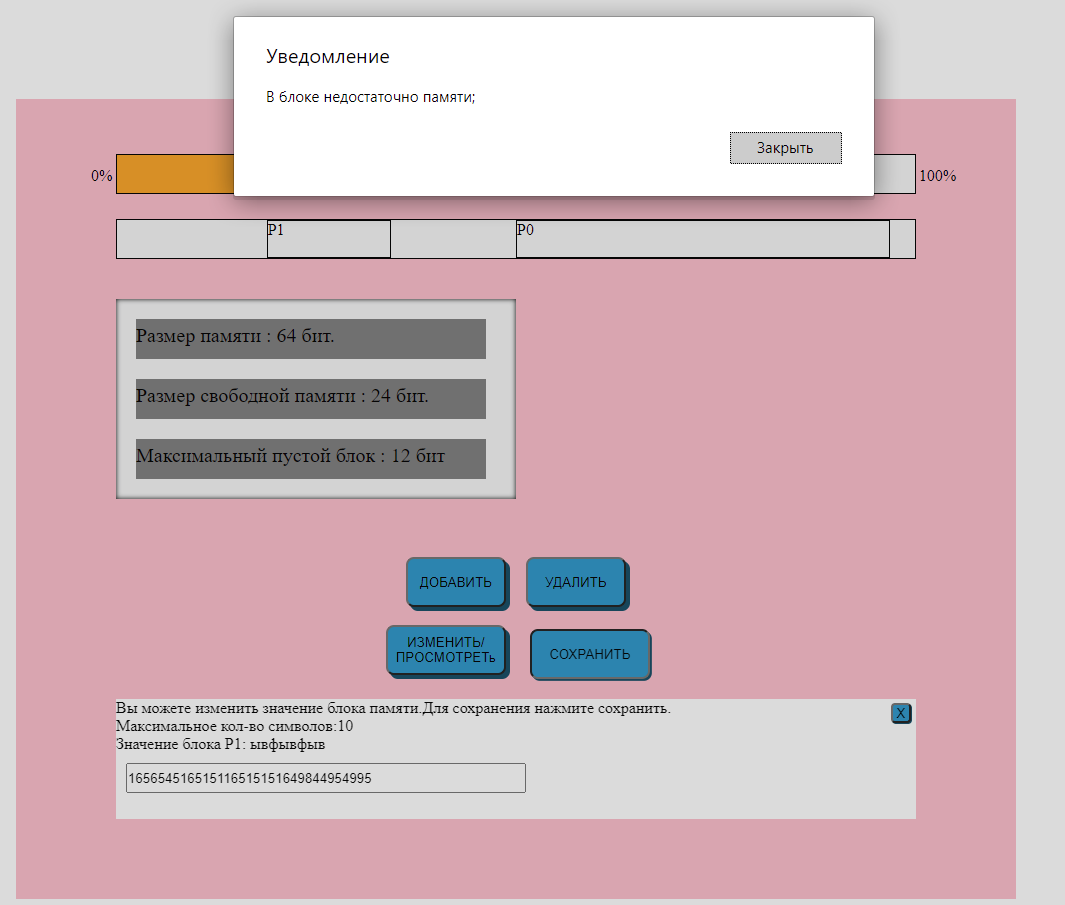


Рисунок 4 – Сообщение об ошибке при попытке записи значение большего чем размер блока

**Вывод**: в ходе работы была реализована вспомогательная утилита «Управление доступом».

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>LAB9</title>

<link rel="stylesheet" type="text/css" href="LAB11.css">

</head>

<body>

<script type="text/javascript">

var n = parseInt(prompt("Введите размер памяти",""));

// var n=1024;

</script>

<div class="container">

<div class="lab9">

<div class="memcont">

<div class="memory proc">

<div id="full"><p class="procent" id="textP"></p></div>

</div>

<p id="leftInf">0%</p><p id="rightInf">100%</p>

</div>

<div class="bloks memory" id="blocks"></div>

<div class="inform">

<div id="volMem" class="inf"></div>

<div id="volFrMem" class="inf"></div>

<div id="maxBlock" class="inf"></div>

</div>

<div class="buttons">

<button id="addBlock" onclick="addBlock()">ДОБАВИТЬ</button>

<button id="removeBlock" onclick="removeBlock()"> УДАЛИТЬ</button>

</div>

<div class="buttonsx">

<button id="writeFILE" onclick="writeFILE()">ИЗМЕНИТЬ/ПРОСМОТРЕТь</button>

<button id="saveFILE" onclick="saveFILE()">СОХРАНИТЬ</button>

</div>

<div id="write">

<p id="writex"></p>

<input type="text" name="" id="inputText" placeholder="Вы можете ввести текст">

<button id="close" onclick="cl()">X</button>

</div>

</div>

</div>

<script type="text/javascript">

document.getElementById('volMem').innerHTML = 'Размер памяти : ' + n + ' бит.';

let arr = new Array(n);

let blockList = new Array(100);

let poss=[];

let possMax = new Array();

let numOfblocks=0;

let glob;

const blocks = document.getElementById('blocks');

for (var i = 0; i < blockList.length; i++) {

blockList[i] = new Array(4);

}

for (var i = 0; i < arr.length; i++) {

arr[i]=0;

}

function chekСompl() {

let k=0;

for (var i = 0; i < arr.length; i++)

if (arr[i]==1) {k++;}

return k/n;

}

function fiendMax() {

let counter=0,max=0;

for (var i = 0; i < arr.length; i++) {

if (arr[i]==0) {counter++;}

if (max<counter) {max=counter;}

else if (arr[i]==1) {counter=0;}

}

return max;

}

function changeInf() {

document.getElementById('volMem').innerHTML = 'Размер памяти : ' + n + ' бит.';

document.getElementById('volFrMem').innerHTML = 'Размер свободной памяти : ' + (n-n\*chekСompl()) + ' бит.';

document.getElementById('maxBlock').innerHTML = 'Максимальный пустой блок : ' + fiendMax() + ' бит';

let k=Math.round(chekСompl()\*100);

const full = document.getElementById('full');

if (k<34) {full.style.backgroundColor= 'green';}

else if (k<67) {full.style.backgroundColor= 'orange';}

else if (k<101) {full.style.backgroundColor= 'red';}

full.style.width = k;

if (k!=0) {k+='%';}

else {k='';}

document.getElementById('textP').innerHTML = k ;

}

function randInt(max) {

return Math.floor(Math.random() \* max);

}

function addBlock() {

let size = parseInt(prompt("Введите размер блока",""));

poss.splice(0,poss.length);

let counter=0;

for (var i = 0; i < arr.length; i++) {

if (arr[i]==0) {counter++;}

if (counter>=size) {

let sizeBlock=i-size;

poss.push(sizeBlock);

}

if (arr[i]==1) {counter=0;}

}

if (poss.length!=0) {

let p = poss[randInt(poss.length)];

let wid = size/n\*100;

wid+='%';

let left = p/n\*100;

left+='%';

for (var i = p ; i < p+size; i++) {

arr[i]=1;

}

let div = document.createElement('div');

div.id ='P'+numOfblocks;

div.className = 'block';

div.innerHTML +='P'+numOfblocks;

div.style.width= wid;

div.style.marginLeft = left;

blocks.appendChild(div);

blockList[numOfblocks][0]='P'+numOfblocks;

blockList[numOfblocks][1]=size;

blockList[numOfblocks][2]=p;

blockList[numOfblocks][3]='';

numOfblocks++;

changeInf();

}

else{alert('Недостаточно памяти!');}

}

function removeBlock() {

let num = parseInt(prompt("Введите номер блока (если p1 введите 1)",""));

let size = blockList[num][1];

let p = blockList[num][2];

let idProc = 'P' + num;

for (var i = p ; i < p+size; i++) {

arr[i]=0;

}

document.getElementById(idProc).remove();

blockList[num][0]=0;

blockList[num][1]=0;

blockList[num][2]=0;

blockList[num][3]='';

changeInf();

}

function writeFILE() {

glob = parseInt(prompt("Введите номер процесса (если p1 введите 1)",""));

const inputText = document.getElementById('inputText');

document.getElementById('saveFILE').style.display = 'block';

document.getElementById('writeFILE').style.marginLeft= '-130px';

document.getElementById('write').style.display = 'block';

document.getElementById('writex').innerHTML = 'Вы можете изменить значение блока памяти.Для сохранения нажмите сохранить.<br> Максимальное кол-во символов:' + blockList[glob][1] + '<br>Значение блока P' + glob + ': ' + blockList[glob][3];

}

function saveFILE() {

const inputText = document.getElementById('inputText');

if (inputText.value.length>blockList[glob][1]) {alert('В блоке недостаточно памяти;');

return false;}

blockList[glob][3] = '' + inputText.value;

inputText.value = '';

document.getElementById('saveFILE').style.display = 'none';

document.getElementById('writeFILE').style.marginLeft= '-60px';

document.getElementById('write').style.display = 'none';

}

function cl() {

document.getElementById('saveFILE').style.display = 'none';

document.getElementById('writeFILE').style.marginLeft= '-60px';

document.getElementById('write').style.display = 'none';

}

changeInf();

</script>

</body>

</html>

**CCS**

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

div.container {

width: 100%;

height:auto;

background-color: black;

position: relative;

}

div.lab9{

position: absolute;

left: 50%;

width: 1000px;

margin-left: -500px;

margin-top: 100px;

background-color: pink;

height: 800px;

}

div.memory {

position: absolute;

left: 50%;

width: 800px;

margin-left: -400px;

height: 40px;

background-color: #f6f6f6;

border: 1px solid;

}

div.memcont {

position: absolute;

left: 50%;

width: 900px;

margin-left: -450px;

margin-top: 50px;

height: 70px;

}

#leftInf {

float: left;

margin: 18px 0 0 25px;

}

#rightInf {

float: right;

margin: 18px 10px 0 0;

}

.proc {

margin-top: 5px;

}

.bloks {

margin-top: 120px;

}

#full {

background-color: green;

height: 100%;

width: 10%;

}

.inform {

position: absolute;

left: 100px;

width: 400px;

margin-top: 200px;

background-color: blue;

height: 200px;

box-shadow: 0 0 5px black inset;

background-color: #f6f6f6;

}

.inf {

margin: 20px;

background-color: gray;

width: 350px;

height: 40px;

font-size: 20px;

padding-top: 5px;

}

.buttons {

position: absolute;

left: 50%;

width: 900px;

margin-left: -450px;

margin-top: 450px;

height: 70px;

}

.buttons button {

position: absolute;

left: 50%;

width: 100px;

height: 50px;

margin-top: 8px;

background-color: #1197cc;

box-shadow: #094c66 4px 4px 0px;

border-radius: 8px;

transition: transform 200ms, box-shadow 200ms;

}

.buttons button:active {

transform: translateY(4px) translateX(4px);

box-shadow: #094c66 0px 0px 0px;

}

#removeBlock {

margin-left: 10px;

}

#addBlock {

margin-left: -110px;

}

.block {

height: 100%;

border: 1px solid;

position: absolute;

}

.procent {

float: right;

margin-top: 12px;

}

.buttonsx {

position: absolute;

left: 50%;

width: 900px;

margin-left: -450px;

margin-top: 518px;

height: 150px;

}

.buttonsx button {

position: absolute;

left: 50%;

width: 120px;

height: 50px;

margin-top: 8px;

background-color: #1197cc;

box-shadow: #094c66 4px 4px 0px;

border-radius: 8px;

transition: transform 200ms, box-shadow 200ms;

}

.buttonsx button:active {

transform: translateY(4px) translateX(4px);

box-shadow: #094c66 0px 0px 0px;

}

#writeFILE{

margin-left: -60px;

}

#saveFILE {

margin-left: 10px;

display: none;

}

#write {

position: absolute;

left: 50%;

width: 800px;

margin-left: -400px;

margin-top: 600px;

height: 120px;

background-color: white;

display: none;

}

input {

width: 400px;

margin: 10px;

height: 30px;

}

#close {

float: right;

width: 20px;

height: 20px;

background-color: #1197cc;

box-shadow: #094c66 1px 1px 0px;

border-radius: 4px;

transition: transform 200ms, box-shadow 200ms;

margin: -50px 5px 0 0;

}

#buttonsx button:active {

transform: translateY(4px) translateX(4px);

box-shadow: #094c66 0px 0px 0px;

}