

«

»

“ ”

“ ”

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Управление ресурсами в вычислительных системах**

: 02.03.03

, :

: 3, : 6

		6
1	()	4
2		144
3	, .	63
4	, .	18
5	, .	0
6	, .	36
7	, .	12
8	, .	2
9	, .	7
10	, .	81
11	(, ,)	
12		

(): 02.03.03

222 12.03.2015 ., : 07.04.2015 .

: 1,

(): 02.03.03

, 4 20.06.2017

, 6 21.06.2017

:

,

:

,

:

.

1.

1.1

Компетенция ФГОС: ОПК.8 способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения (далее - ПО; в части следующих результатов обучения):	
10.	
9.	-
10.	
7.	
Компетенция НГТУ: ПК.10.В/ЭУ готовность к использованию современных операционных систем, оболочек и сервисных программ; в части следующих результатов обучения:	
4.	
5.	

2.

2.1

--	--

.10. / . 4	
1.Архитектуру операционной системы Unix	; ;
2.Архитектуру операционной системы Windows	; ;
.10. / . 5	
3.Основы разработки программного обеспечения	; ;
.8. 9	
4.Основы разработки клиент-серверного взаимодействия	; ;
.8. 10	
5.Управления вычислительными ресурсами	; ;
.8. 7	
6.Использования современных программных средств низкого уровня в задачах управления вычислительными ресурсами	; ;
.8. 10	
7.Методы проектировать программное обеспечение низкого уровня	; ;

3.

3.1

: 6						
: UNIX						

<p>7.</p> <p>wait.</p> <p>exec.</p> <p>Kill().</p> <p>Signal().</p>	<p>0</p>	<p>1</p>	<p>1, 3, 5, 6, 7</p>	
<p>8.</p> <p>Shell.</p> <p>Shell.</p>	<p>0</p>	<p>0,5</p>	<p>1, 3, 5, 7</p>	
<p>IPC</p>				
<p>9.</p> <p>V-</p> <p>P-</p> <p>P-, V-</p>	<p>0</p>	<p>1</p>	<p>1, 3, 5, 6, 7</p>	

<p>10. IPC Unix. IPC.</p>	<p>0</p>	<p>1</p>	<p>1, 3, 5, 6, 7</p>	
:				
<p>11. -</p> <p>IP-</p>	<p>0</p>	<p>1</p>	<p>1, 3, 4, 5, 6, 7</p>	
<p>12. TCP-</p> <p>(socket). (bind).</p> <p>TCP- (listen).</p> <p>TCP- (accept). (connect). (send, recv).</p> <p>TCP- (close).</p>	<p>0</p>	<p>1</p>	<p>1, 3, 4, 5, 6, 7</p>	

13. (socket). (bind). (bind). (connect). (send, recv). TCP- (close).	0	0,5	1, 3, 4, 5, 6, 7	
14. RPC. Middleware. RPC. RPC.	0	1	1, 3, 4, 5, 6, 7	
:				
15. (demand paging). (thrashing). -stealer. -swapper.	0	1	1, 2	
: Windows				
16. Windows. Windows.	0	1	2, 3, 5, 6, 7	
17. Windows Windows.	0	1	2, 3, 5, 6, 7	

18.	Windows.	0	1	2, 3, 5, 6, 7	
19.	Windows.	0	1	2, 3, 5, 6, 7	

3.2

: 6					
: UNIX					
1.	Shell- Unix.	1,5	4	1, 5	
:					
2.	Unix.	1,5	4	3, 5, 6, 7	
3.	Unix.	1,5	4	3, 5, 6, 7	
:					
4.	Unix.	1,5	4	3, 5, 6, 7	
5.	Unix.	1,5	4	3, 5, 6, 7	
6.	Shell- Unix.	1,5	4	3, 5, 6, 7	
: IPC					

7.	Unix.	1,5	6	3, 5, 6, 7	
:					
8.	TCP-Unix.	1,5	6	3, 5, 6, 7	

4.

: 6					
1			3, 4, 5, 6, 7	20	4
<p>Windows : OC</p> <p>" / . . . - ;[. . .].- , 2012. - 26, [3] ..- :</p> <p>http://elibrary.nstu.ru/source?bib_id=vtls000171001</p>					
2			3, 4, 5, 6, 7	32	0
<p>UNIX.</p> <p>[]: , [2011]. -</p> <p>: http://ami.nstu.ru/~vms/method6/Method1.HTM. -</p>					
3			1, 2, 3, 4, 5, 6, 7	20	3
<p>OC Windows :</p> <p>3 "</p> <p>" / . . . - ;[. . .].- , 2012. - 26, [3] ..- :</p> <p>http://elibrary.nstu.ru/source?bib_id=vtls000171001</p> <p>UNIX. []: , [2011]. -</p> <p>- : http://ami.nstu.ru/~vms/method6/Method1.HTM. -</p>					
4			1, 2, 3, 4, 5, 6, 7	9	0
<p>OC Windows :</p> <p>3 "</p> <p>" / . . . - ;[. . .].- , 2012. - 26, [3] ..- :</p> <p>http://elibrary.nstu.ru/source?bib_id=vtls000171001</p> <p>UNIX. []: , [2011]. -</p> <p>- : http://ami.nstu.ru/~vms/method6/Method1.HTM. -</p>					

5.

(. 5.1).

5.1

	e-mail; ;
	e-mail;
	;

5.2

1		.8; .10. /
<p>Формируемые умения: з10. Знать основы управления современными вычислительными ресурсами; з4. Знать основы управления вычислительными сетями; з5. Знать основные подходы в области системного моделирования; у10. Уметь проектировать программное обеспечение низкого уровня; у7. Уметь использовать современные программные средства низкого уровня в задачах управления вычислительными ресурсами</p>		
<p>Краткое описание применения: Обсуждение выполненных программных разработок на лабораторных занятиях</p>		
<p>UNIX. []: ; , [2011]. - : http://ami.nstu.ru/~vms/method6/Method1.HTM. - ."</p>		

6.

(),

15-

ECTS.

. 6.1.

1

6.1

: 6		
Лабораторная:	28	42
<p>() " UNIX. []: ; , [2011]. - : http://ami.nstu.ru/~vms/method6/Method1.HTM. - ."</p>		
РГЗ:	4	18
<p>() " OC Windows : 3 " "/ ; []: . , 2012. - 26, [3] . : : http://elibrary.nstu.ru/source?bib_id=vtls000171001</p>		
Экзамен: Выполнение тестовых заданий на лекции	0	24
<p>() " UNIX. []: ; , [2011]. - : http://ami.nstu.ru/~vms/method6/Method1.HTM. - ."</p>		
Экзамен: Выполнение заданий на экзамене	0	16

6.2

6.2

		/		
.8	10.	+	+	+
	9.	+		+
	10.	+	+	+
	7.	+		+
	.10. / 4.	+	+	+
	.10. / 5.	+		+

1

7.

- Олифер В. Г. Сетевые операционные системы : учебное пособие для вузов по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / В. Г. Олифер, Н. А. Олифер. - СПб. [и др.], 2007. - 538 с. : ил.
- Стивенс У. Р. UNIX. Разработка сетевых приложений / У. Р. Стивенс, Б. Феннер, Э. М. Рудофф. - СПб., 2007. - 1038 с. : ил.
- Стивенс У. Р. UNIX. Профессиональное программирование / У. Ричард Стивенс, Стивен А. Раго ; [пер. А. Киселева]. - СПб. ;, 2007. - 1035 с. : ил.
- Стасьшин В. М. Управление ресурсами в ОС Windows : конспект лекций / В. М. Стасьшин ; Новосиб. гос. техн. ун-т. - Новосибирск, 2013. - 100, [3] с. : ил., табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000182617

- Робачевский А. М. Операционная система UNIX. - СПб., 2000. - 514 с. : ил.
- Таненбаум Э. С. Операционные системы. Разработка и реализация / Э. Таненбаум, А. Вудхалл. - СПб., 2007. - 702 с. : ил. + 1 CD-ROM.
- Моли Б. Unix/Linux. Теория и практика программирования / Брюс Моли ; пер. с англ. [В. Д. Никитина]. - М., 2004. - 576 с.
- Богачев К. Ю. Основы параллельного программирования / К. Ю. Богачёв. - М., 2003. - 342 с.
- Хэвиленд К. Системное программирование в UNIX : Руководство программиста по разработке ПО / К. Хэвиленд, Д. Грэй, Б. Салама. - М., 2000. - 364 с.
- Дансмур М. Операционная система UNIX и программирование на языке СИ / М. Дансмур, Г. Дейвис; Пер. с англ. А. С. Богданова, под ред. И. Г. Шестакова. - М., 1989. - 192 с.
- Джордейн Р. Справочник программиста персональных компьютеров типа IBM PC, XT и AT / Р. Джордейн ; пер. с англ. Н. В. Гайского. - М., 1991. - 543 с. : ил.

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniium.com" : <http://znaniium.com/>
5. :

8.

8.1

1. Стасьшин В. М. Управление ресурсами в ОС UNIX. Методические указания для выполнения лабораторных работ [Электронный ресурс] : учебно-методическое пособие / В. М. Стасьшин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: <http://ami.nstu.ru/~vms/method6/Method1.HTM>. - Загл. с экрана.
2. Управление ресурсами в ОС Windows : методические указания к выполнению индивидуальной работы для 3 курса ФПМИ по курсу "Управление ресурсами в вычислительных системах" / Новосиб. гос. техн. ун-т ; [сост.: В. М. Стасьшин, И. М. Куликов]. - Новосибирск, 2012. - 26, [3] с. : табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000171001

8.2

- 1 Операционные системы семейства LINUX
- 2 Microsoft Windows

9.

-

1	(- , ,)	
2	(Internet)	

Правила аттестации студентов по учебной дисциплине “Управление ресурсами в вычислительных системах”

1. Оценка работы студента в течение семестра.

Работа студента по освоению учебной дисциплины в течение семестра складывается из

- изучения теоретического материала учебной дисциплины в соответствии с рабочей программой;
- выполнения лабораторных работ по курсу;
- выполнения расчетно-графической работы (РГР) по курсу.

При оценке выполнения лабораторных работ учитываются:

- правильность написания программы или скрипта на языке Shell-интерпретатора при выполнении варианта задания;
- понимание логики программы, а также теоретического материала, на основании которого эта программа написана;
- уровень документируемости программы;
- достаточность и качество предлагаемых тестов;
- качество оформления отчета по лабораторной работе;
- соответствие графику выполнения лабораторных работ;
- ответы на вопросы по лабораторной работе.

Максимальное количество баллов за выполнение каждой из работ – от 4 до 6 в зависимости от сложности лабораторной работы.

Приём каждой лабораторной работы включает 3 этапа:

- 1) демонстрация работы программы на компьютере (проверяется правильность работы программы);
- 2) объяснение содержимого отчёта (проверяется понимание логики работы программы, знание теоретического материала, на основе которого написана программа, соответствие отчёта установленным требованиям);
- 3) защита лабораторной работы (ответы на контрольные вопросы).

Задание по РГР выполняется индивидуально каждым студентом. Разрабатываемая программа должна удовлетворять требованиям I, II, III или IV уровня. Удовлетворение требованиям I уровня является обязательным для всех студентов. Соблюдение требований II, III или IV уровня обязательным не является.

Для проверки выполненной индивидуальной работы студент по электронной почте отправляет преподавателю, осуществляющему проверку выполнения индивидуальной работы, на адрес его электронной почты письмо, содержащее комплект файлов с результатами выполнения индивидуальной работы. Требования к оформлению письма и составу файлов содержатся в методических указаниях по выполнению РГР. Кроме этого студент предоставляет в печатном виде отчет о выполнении РГР.

При оценке выполнения исследовательской работы учитываются:

- правильность написания программы для указанного варианта задания;
- понимание логики программы, а также теоретического материала, на основании которого эта программа написана;
- уровень документируемости программы;
- достаточность и качество предлагаемых тестов;
- качество оформления отчета по РГР.

Результаты проверки РГР преподаватель выставляет на своей странице на портале НГТУ.

Все лабораторные работы выполняются в течение семестра, выполнение РГР ведется параллельно с выполнением лабораторных работ.

Перечень работ в семестре, срок их выполнения и оценка в баллах представлены таблицей:

Темы лабораторных работ	Срок выполнения по графику (неделя)	Максимальная оценка в баллах	Минимальная оценка в баллах
1. Лабораторные работы по курсу			
• Лабораторная работа №1. Управление системными ресурсами Shell-интерпретатора.	1-4	6	4
• Лабораторная работа №2. Файловая система ОС UNIX.	1-4	5	3
• Лабораторная работа №3. Структура системы управления вводом-выводом в ОС UNIX.	5-6	4	2
• Лабораторная работа №4. Порождение нового процесса и работа с ним в рамках одной программы. Запуск программы в рамках порожденного процесса. Сигналы и каналы в ОС UNIX.	7-8	5	3
• Лабораторная работа №5. Синхронизация процессов.	9-10	5	4
• Лабораторная работа №6. Моделирование работы интерпретатора.	11-12	5	4
• Лабораторная работа №7. Клиент-серверные взаимодействия посредством сокетов в режиме TCP-соединения.	13-15	6	4
• Лабораторная работа №8. Межпроцессное взаимодействие программ.	16-18	6	4
2. Выполнение индивидуальной исследовательской работы (РГР) по теме “Управление ресурсами в ОС Windows”			
I уровень	12-18	7	4
II уровень		11	6
III уровень		15	8
IV уровень		18	10

Опережение сроков выполнения лабораторных работ и РГР приветствуется.

Максимальное количество баллов, набранное за выполнение лабораторных работ и индивидуальной исследовательской работы (РГР) – 60.

Минимальное количество баллов, набранное за выполнение лабораторных работ и индивидуальной исследовательской работы (РГР) – 32.

Замечание. За выполнение учебных заданий сверх предусмотренных основной программой освоения дисциплины (учебно-исследовательская работа студента по тематике, смежной с тематикой учебного курса, самостоятельное углубленное освоение отдельных тем, участие в предметных олимпиадах различного уровня и пр.) студенту могут быть добавлены дополнительные баллы (не более 20).

2. Проверка уровня знаний студента

Проверка уровня знаний студента проводится как в течение семестра, так и в форме итогового экзамена.

2.1. Проверка знаний отдельных тем учебной дисциплины в течение семестра.

Проверка знаний отдельных тем учебной дисциплины в течение семестра осуществляется в тестовой форме (6 тестов).

Количество баллов, набранных за выполнение одного теста, рассчитывается по формуле:

$$(4,0 * \text{Процент правильных ответов}) / 100$$

Максимальное количество баллов, набранное за выполнение одного теста – 4,0.

Максимальное количество баллов, набранное за выполнение шести тестов – 24.

2.2. Итоговая аттестация на экзамене.

Для допуска к экзамену студент должен набрать в течение семестра **не менее 35 баллов**, при этом

- **за выполнение лабораторных работ студент должен набрать не менее 20 баллов,**
- **должна быть выполнена индивидуальная исследовательская работа с одним из уровней сложности.**

Итоговая аттестация на экзамене включает:

- написание программы по тематике лабораторных работ учебной дисциплины;
- ответы на вопросы по содержанию учебной дисциплины.

Максимальное количество баллов, полученных на экзамене – 16.

Максимальное количество баллов, полученных при проверке знаний студента – 40, из них

**в течение семестра – 24,
во время экзамена – 16.**

Если по результатам работы в семестре (выполнение лабораторных работ, РГР, учебных заданий сверх предусмотренных основной программой освоения дисциплины) и аттестации студента в течение семестра (проверка знаний при выполнении тестов) студент набрал более 90 баллов, итоговая оценка может быть выставлена без проведения экзамена (“автомат”) с проставлением в ведомость и зачетную книжку оценки “Отлично”, что соответствует группе уровней “А” по шкале ECTS.

3. Повторная аттестация по курсу.

Повторная аттестация (в случае получения студентом оценки FX) проводится в сроки, определенные «Положением об экзаменационной сессии» по направлению деканата.

Для допуска к повторной аттестации студент должен **выполнить и сдать расчетно-графическую работу с одним из уровней сложности, выполнить (досдать) лабораторные работы, набрав (с учетом тестов, выполненных в течение семестра) не менее 40 баллов.**

При прохождении повторной аттестации студент отвечает на вопросы по содержанию учебной дисциплины;

В случае повторной аттестации итоговая оценка не может быть выше «Е».

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Управление ресурсами в вычислительных системах приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.8 способность использовать знания методов проектирования и производства программного продукта, принцип построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения (далее - ПО	з9. Знать основы разработки клиент-серверного взаимодействия	Клиент-серверные взаимодействия. Сервер. Клиент. Архитектура клиент-сервер. Сервисы. IP-адрес. Понятие серверного порта. Модель соединений. Модель дейтаграмм. Обобщенные сокеты. Сокеты для связи через сеть. Адресация в соquete. Дескриптор сокета. Создание сокета. Программирование в режиме TCP-соединения. Структура серверного процесса. Структура клиентского процесса (socket). Связывание (bind). Включение приема TCP-соединений (listen). Прием запроса на установку TCP-соединений (accept). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Программирование в режиме дейтаграмм. Структура серверного процесса. Структура клиентского процесса (socket). Связывание сервера с сокетом (bind). Связывание клиента с сокетом (bind). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Удаленный вызов процедур RPC. Категории продуктов Middleware. Неявное сетевое программирование. Технология использования RPC. Схема функционирования RPC.	Отчет по лабораторной работе № 8, вопросы к лабораторным работам.	Экзамен, вопросы 19-22. Вопросы теста №6
ОПК.8	з10. Знать основы управления современными вычислительными ресурсами	Ресурсы ОС Unix. Функции подсистемы управления памятью. Функции подсистемы управления процессами. Функции подсистемы управления устройствами. Структура процесса. Таблица процессов. Промежуточная таблица процессов. Таблица областей. Контекст процесса. Информационное взаимодействие между процессами. Структура элемента таблицы процессов. U-область. Состояния процессов. Диаграмма состояний. Каналы. Наследование пользовательских дескрипторов. Понятие канала. Виды каналов. Средства организации непоименованных каналов. Алгоритм pipe. Создание поименованного канала. Чтение и запись через поименованный и непоименованный каналы. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при работе с каналом. Наследование пользовательских дескрипторов. Системный вызов Dup(). Системный вызов Fcntl(). Переназначение стандартного ввода и стандартного вывода. Клиент-серверные взаимодействия. Сервер. Клиент. Архитектура клиент-сервер. Сервисы. IP-адрес. Понятие серверного порта. Модель соединений. Модель дейтаграмм. Обобщенные сокеты. Сокеты для связи через сеть. Адресация в соquete. Дескриптор сокета. Создание сокета. Командный процессор	Отчет по лабораторной работе (лабораторные работы 1-8) , вопросы к лабораторным работам. Отчет по РГЗ	Экзамен, вопросы 1-40. Вопросы тестов №1-6

Shell. Использование программного канала для передачи данных между процессами. Функциональная схема построения конвейера. Основной цикл программы Shell. Системные операции, связанные со временем. Написание программы на языке, моделирующей работу Shell-интерпретатора ОС Unix. Написание программы на языке Си для осуществления клиент-серверные взаимодействия посредством сокетов в режиме TCP-соединения в ОС Unix. Написание программы на языке Си для, осуществляющих межпроцессное взаимодействие основе семафоров, очередей сообщений и разделяемой пвмяти в ОС Unix. Написание программы на языке Си для синхронизации процессов посредством сигналов и каналов ОС Unix. Написание программы на языке Си для управления процессами ОС Unix. Написание программы на языке Си для управления ресурсами файловой системы ОС Unix. Написание программы на языке Си, моделирующих состояние системных таблиц ядра системы ввода-вывода ОС Unix. Написание скриптов на языке ресурсами Shell-интерпретатора для управления ресурсами ОС Unix. Обмен данными между параллельными процессами в ОС Windows. Способы передачи данных между процессами. Синхронный и асинхронный обмен данными. Работа с неименованными (анонимными) каналами. Работа с именованными каналами. Основные понятия ОС Unix. Процесс. Программа. Ядро. Функции ядра. Режим ядра и режим пользователя. Утилиты. Критическая секция. Системные вызовы. Схема обработки системных вызовов. Прерывания и их приоритеты. Мультипрограммирование и своппинг. Диспетчерский процесс. Общие аспекты управления памятью. Виртуальная и физическая память (1, 2, 7, 8). Основы теории семафоров. Понятие критической секции. Двоичный семафор. Общий семафор. Примитивы для работы с семафорами. P-операция. V-операция. Примеры использования P-, V-операций в задаче взаимного исключения. Использование семафоров в алгоритме банкира. Программирование в режиме TCP-соединения. Структура серверного процесса. Структура клиентского процесса (socket). Связывание (bind). Включение приема TCP-соединений (listen). Прием запроса на установку TCP-соединений (accept). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Программирование в режиме дейтаграмм. Структура серверного процесса. Структура клиентского процесса (socket). Связывание сервера с сокетом (bind). Связывание клиента с сокетом (bind). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Синхронизация процессов и потоков в ОС Windows. Критические секции. Примитивы синхронизации. События. Семафоры. Программная и аппаратная реализация синхронизации. Тупики. Обнаружение и предотвращение тупиков. Синхронизация процессов. Приостановление выполнения процесса. Алгоритм wait. Вызов программы в рамках порожденного процесса. Алгоритм exes. Сигналы. Взаимодействие процессов посредством сигналов. Моменты проверки и обработки сигналов на диаграмме состояний. Системный вызов Kill(). Алгоритм обработки сигналов. Системный вызов Signal(). Возможность побочных эффектов. Создание и завершение процесса. Понятие процесса. Идентификатор процесса. Наследование контекста среды при создании процесса. Системный вызов fork(). Группа процессов. Алгоритм fork. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при создании процесса. Способы завершения

		<p>процесса. Алгоритм exit. Разделение родительским и порожденным процессами доступа к файлу. Взаимодействие процессов посредством каналов. Разделение процессами канала. Средства IPC в ОС Unix. Общие понятия средств IPC. Ключ. Идентификатор. Структура статуса межпроцессового взаимодействия. Синхронизация на основе семафоров. Наборы семафоров. Создание семафора. Инициализация семафора. Операции с семафорами. Очереди сообщений. Создание очереди. Структуры данных, используемые для организации очередей сообщений. Помещение и выборка сообщений из очереди. Разделяемая память. Создание сегмента разделяемой памяти. Подключение сегмента к виртуальной памяти процесса. Взаимодействие процессов посредством механизма IPC. Удаленный вызов процедур RPC. Категории продуктов Middleware. Неявное сетевое программирование. Технология использования RPC. Схема функционирования RPC. Управление потоками в ОС Windows. Потоки в ОС Windows. Контекст потока. Состояние потока. Диспетчеризация и планирование потоков. Создание и завершение потока. Приостановка и возобновление потока. Управление процессами в ОС Windows. Процессы в ОС Windows. Создание и завершение процесса. Наследование дескрипторов. Дублирование дескрипторов. Обслуживание потоков. Управление файлами. Системные операции при работе с файловой системой. Алгоритм Open. Алгоритм Creat. Чтение и запись в файл. Алгоритм Read. Движение по файлу. Алгоритм Close. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при выполнении системных операций над файлами. Системные вызовы для работы с каталогами. Дополнительные системные вызовы при работе с файловой системой. Элементы файловой системы. Структура файловой системы. Структура индексного дескриптора. Суперблок. Выделение и освобождение дисковых блоков. Назначение описателя файла новому файлу. Таблица описателей файлов. Таблица файлов. Таблица открытых файлов.</p>		
ОПК.8	у7. Уметь использовать современные программные средства низкого уровня в задачах управления вычислительными ресурсами	<p>Клиент-серверные взаимодействия. Сервер. Клиент. Архитектура клиент-сервер. Сервисы. IP-адрес. Понятие серверного порта. Модель соединений. Модель дейтаграмм. Обобщенные сокеты. Сокеты для связи через сеть. Адресация в соquete. Дескриптор сокета. Создание сокета. Написание программы на языке, моделирующей работу Shell-интерпретатора ОС Unix. Написание программы на языке Си для осуществления клиент-серверные взаимодействия посредством сокетов в режиме TCP-соединения в ОС Unix. Написание программы на языке Си для, осуществляющих межпроцессное взаимодействие основе семафоров, очередей сообщений и разделяемой пвмяти в ОС Unix. Написание программы на языке Си для синхронизации процессов посредством сигналов и каналов ОС Unix. Написание программы на языке Си для управления процессами ОС Unix. Написание программы на языке Си для управления ресурсами файловой системы ОС Unix. Написание программы на языке Си, моделирующих состояние системных таблиц ядра системы ввода-вывода ОС Unix. Обмен данными между параллельными процессами в ОС Windows. Способы передачи данных между процессами. Синхронный и асинхронный обмен данными. Работа с именованными (анонимными) каналами. Работа с именованными каналами. Основы теории семафоров. Понятие критической секции. Двоичный семафор. Общий семафор. Примитивы для работы с семафорами. Р-</p>	Отчет по лабораторной работе (лабораторные работы 1-8) , вопросы к лабораторным работам. Отчет по РГЗ	Экзамен, вопросы 1-40. Вопросы тестов №1-6

		<p>операция. V-операция. Примеры использования P-, V-операций в задаче взаимного исключения. Использование семафоров в алгоритме банкира. Программирование в режиме TCP-соединения. Структура серверного процесса. Структура клиентского процесса (socket). Связывание (bind). Включение приема TCP-соединений (listen). Прием запроса на установку TCP-соединений (accept). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Программирование в режиме дейтаграмм. Структура серверного процесса. Структура клиентского процесса (socket). Связывание сервера с сокетом (bind). Связывание клиента с сокетом (bind). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Синхронизация процессов и потоков в ОС Windows. Критические секции. Примитивы синхронизации. События. Семафоры. Программная и аппаратная реализация синхронизации. Тупики. Обнаружение и предотвращение тупиков. Синхронизация процессов. Приостановление выполнения процесса. Алгоритм wait. Вызов программы в рамках порожденного процесса. Алгоритм exec. Сигналы. Взаимодействие процессов посредством сигналов. Моменты проверки и обработки сигналов на диаграмме состояний. Системный вызов Kill(). Алгоритм обработки сигналов. Системный вызов Signal(). Возможность побочных эффектов. Создание и завершение процесса. Понятие процесса. Идентификатор процесса. Наследование контекста среды при создании процесса. Системный вызов fork(). Группа процессов. Алгоритм fork. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при создании процесса. Способы завершения процесса. Алгоритм exit. Разделение родительским и порожденным процессами доступа к файлу. Взаимодействие процессов посредством каналов. Разделение процессами канала. Средства IPC в ОС Unix. Общие понятия средств IPC. Ключ. Идентификатор. Структура статуса межпроцессового взаимодействия. Синхронизация на основе семафоров. Наборы семафоров. Создание семафора. Инициализация семафора. Операции с семафорами. Очереди сообщений. Создание очереди. Структуры данных, используемые для организации очередей сообщений. Помещение и выборка сообщений из очереди. Разделяемая память. Создание сегмента разделяемой памяти. Подключение сегмента к виртуальной памяти процесса. Взаимодействие процессов посредством механизма IPC. Удаленный вызов процедур RPC. Категории продуктов Middleware. Неявное сетевое программирование. Технология использования RPC. Схема функционирования RPC. Управление потоками в ОС Windows. Потоки в ОС Windows. Контекст потока. Состояние потока. Диспетчеризация и планирование потоков. Создание и завершение потока. Приостановка и возобновление потока. Управление процессами в ОС Windows. Процессы в ОС Windows. Создание и завершение процесса. Наследование дескрипторов. Дублирование дескрипторов. Обслуживание потоков. Элементы файловой системы. Структура файловой системы. Структура индексного дескриптора. Суперблок. Выделение и освобождение дисковых блоков. Назначение описателя файла новому файлу. Таблица описателей файлов. Таблица файлов. Таблица открытых файлов.</p>		
--	--	---	--	--

ОПК.8	у10. Уметь проектировать программное обеспечение низкого уровня	<p>Каналы. Наследование пользовательских дескрипторов. Понятие канала. Виды каналов. Средства организации непоименованных каналов. Алгоритм pipe. Создание поименованного канала. Чтение и запись через поименованный и непоименованный каналы. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при работе с каналом. Наследование пользовательских дескрипторов. Системный вызов Dup(). Системный вызов Fcntl(). Переназначение стандартного ввода и стандартного вывода. Клиент-серверные взаимодействия. Сервер. Клиент. Архитектура клиент-сервер. Сервисы. IP-адрес. Понятие серверного порта. Модель соединений. Модель дейтаграмм. Обобщенные сокеты. Сокеты для связи через сеть. Адресация в сокете. Дескриптор сокета. Создание сокета. Командный процессор Shell. Использование программного канала для передачи данных между процессами. Функциональная схема построения конвейера. Основной цикл программы Shell. Системные операции, связанные со временем. Написание программы на языке, моделирующей работу Shell-интерпретатора ОС Unix. Написание программы на языке Си для осуществления клиент-серверных взаимодействий посредством сокетов в режиме TCP-соединения в ОС Unix. Написание программы на языке Си для, осуществляющих межпроцессное взаимодействие основе семафоров, очередей сообщений и разделяемой пвмяти в ОС Unix. Написание программы на языке Си для синхронизации процессов посредством сигналов и каналов ОС Unix. Написание программы на языке Си для управления процессами ОС Unix. Написание программы на языке Си для управления ресурсами файловой системы ОС Unix. Написание программы на языке Си, моделирующих состояние системных таблиц ядра системы ввода-вывода ОС Unix. Обмен данными между параллельными процессами в ОС Windows. Способы передачи данных между процессами. Синхронный и асинхронный обмен данными. Работа с неименованными (анонимными) каналами. Работа с именованными каналами. Основы теории семафоров. Понятие критической секции. Двоичный семафор. Общий семафор. Примитивы для работы с семафорами. P-операция. V-операция. Примеры использования P-, V-операций в задаче взаимного исключения. Использование семафоров в алгоритме банкира. Программирование в режиме TCP-соединения. Структура серверного процесса. Структура клиентского процесса (socket). Связывание (bind). Включение приема TCP-соединений (listen). Прием запроса на установку TCP-соединений (accept). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Программирование в режиме дейтаграмм. Структура серверного процесса. Структура клиентского процесса (socket). Связывание сервера с сокетом (bind). Связывание клиента с сокетом (bind). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Синхронизация процессов и потоков в ОС Windows. Критические секции. Примитивы синхронизации. События. Семафоры. Программная и аппаратная реализация синхронизации. Тупики. Обнаружение и предотвращение тупиков. Синхронизация процессов. Приостановление выполнения процесса. Алгоритм wait. Вызов программы в рамках порожденного процесса. Алгоритм exec. Сигналы. Взаимодействие процессов посредством сигналов. Моменты проверки и обработки сигналов на диаграмме состояний. Системный вызов Kill(). Алгоритм обработки сигналов. Системный вызов Signal(). Возможность побочных</p>	Отчет по лабораторной работе (лабораторные работы 1-8) , вопросы к лабораторным работам. Отчет по РГЗ	Экзамен, вопросы 1-40. Вопросы тестов №1-6
-------	---	--	---	--

		<p>эффектов. Создание и завершение процесса. Понятие процесса. Идентификатор процесса. Наследование контекста среды при создании процесса. Системный вызов fork(). Группа процессов. Алгоритм fork. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при создании процесса. Способы завершения процесса. Алгоритм exit. Разделение родительским и порожденным процессами доступа к файлу. Взаимодействие процессов посредством каналов. Разделение процессами канала. Средства IPC в ОС Unix. Общие понятия средств IPC. Ключ. Идентификатор. Структура статуса межпроцессового взаимодействия. Синхронизация на основе семафоров. Наборы семафоров. Создание семафора. Инициализация семафора. Операции с семафорами. Очереди сообщений. Создание очереди. Структуры данных, используемые для организации очередей сообщений. Помещение и выборка сообщений из очереди. Разделяемая память. Создание сегмента разделяемой памяти. Подключение сегмента к виртуальной памяти процесса. Взаимодействие процессов посредством механизма IPC. Удаленный вызов процедур RPC. Категории продуктов Middleware. Неявное сетевое программирование. Технология использования RPC. Схема функционирования RPC. Управление потоками в ОС Windows. Потоки в ОС Windows. Контекст потока. Состояние потока. Диспетчеризация и планирование потоков. Создание и завершение потока. Приостановка и возобновление потока. Управление процессами в ОС Windows. Процессы в ОС Windows. Создание и завершение процесса. Наследование дескрипторов. Дублирование дескрипторов. Обслуживание потоков. Управление файлами. Системные операции при работе с файловой системой. Алгоритм Open. Алгоритм Creat. Чтение и запись в файл. Алгоритм Read. Движение по файлу. Алгоритм Close. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при выполнении системных операций над файлами. Системные вызовы для работы с каталогами. Дополнительные системные вызовы при работе с файловой системой. Элементы файловой системы. Структура файловой системы. Структура индексного дескриптора. Суперблок. Выделение и освобождение дисковых блоков. Назначение описателя файла новому файлу. Таблица описателей файлов. Таблица файлов. Таблица открытых файлов.</p>		
<p>ПК.10.В/ЭУ готовность к использованию современных операционных систем, оболочек и сервисных программ</p>	<p>34. Знать основы управления вычислительными сетями</p>	<p>Ресурсы ОС Unix. Функции подсистемы управления памятью. Функции подсистемы управления процессами. Функции подсистемы управления устройствами. Структура процесса. Таблица процессов. Промежуточная таблица процессов. Таблица областей. Контекст процесса. Информационное взаимодействие между процессами. Структура элемента таблицы процессов. U-область. Состояния процессов. Диаграмма состояний. Каналы. Наследование пользовательских дескрипторов. Понятие канала. Виды каналов. Средства организации непоименованных каналов. Алгоритм pipe. Создание поименованного канала. Чтение и запись через поименованный и непоименованный каналы. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при работе с каналом. Наследование пользовательских дескрипторов. Системный вызов Dup(). Системный вызов Fcntl(). Переназначение стандартного ввода и стандартного вывода. Клиент-серверные взаимодействия. Сервер. Клиент. Архитектура клиент-сервер. Сервисы. IP-адрес. Понятие серверного порта. Модель соединений. Модель дейтаграмм. Обобщенные сокеты. Сокеты для связи через сеть.</p>	<p>Отчет по лабораторной работе (лабораторные работы 1-8) , вопросы к лабораторным работам. Отчет по РГЗ</p>	<p>Экзамен, вопросы 1-40. Вопросы тестов №1-6</p>

Адресация в сокете. Дескриптор сокета. Создание сокета. Командный процессор Shell. Использование программного канала для передачи данных между процессами. Функциональная схема построения конвейера. Основной цикл программы Shell. Системные операции, связанные со временем. Написание скриптов на языке ресурсами Shell-интерпретатора для управления ресурсами ОС Unix. Обмен данными между параллельными процессами в ОС Windows. Способы передачи данных между процессами. Синхронный и асинхронный обмен данными. Работа с неименованными (анонимными) каналами. Работа с именованными каналами. Основные понятия ОС Unix. Процесс. Программа. Ядро. Функции ядра. Режим ядра и режим пользователя. Утилиты. Критическая секция. Системные вызовы. Схема обработки системных вызовов. Прерывания и их приоритеты. Мультипрограммирование и своппинг. Диспетчерский процесс. Общие аспекты управления памятью. Виртуальная и физическая память (1, 2, 7, 8). Основы теории семафоров. Понятие критической секции. Двоичный семафор. Общий семафор. Примитивы для работы с семафорами. P-операция. V-операция. Примеры использования P-, V-операций в задаче взаимного исключения. Использование семафоров в алгоритме банкира. Программирование в режиме TCP-соединения. Структура серверного процесса. Структура клиентского процесса (socket). Связывание (bind). Включение приема TCP-соединений (listen). Прием запроса на установку TCP-соединений (accept). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Программирование в режиме дейтаграмм. Структура серверного процесса. Структура клиентского процесса (socket). Связывание сервера с сокетом (bind). Связывание клиента с сокетом (bind). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Синхронизация процессов и потоков в ОС Windows. Критические секции. Примитивы синхронизации. События. Семафоры. Программная и аппаратная реализация синхронизации. Тупики. Обнаружение и предотвращение тупиков. Синхронизация процессов. Приостановление выполнения процесса. Алгоритм wait. Вызов программы в рамках порожденного процесса. Алгоритм exec. Сигналы. Взаимодействие процессов посредством сигналов. Моменты проверки и обработки сигналов на диаграмме состояний. Системный вызов Kill(). Алгоритм обработки сигналов. Системный вызов Signal(). Возможность побочных эффектов. Создание и завершение процесса. Понятие процесса. Идентификатор процесса. Наследование контекста среды при создании процесса. Системный вызов fork(). Группа процессов. Алгоритм fork. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при создании процесса. Способы завершения процесса. Алгоритм exit. Разделение родительским и порожденным процессами доступа к файлу. Взаимодействие процессов посредством каналов. Разделение процессами канала. Средства IPC в ОС Unix. Общие понятия средств IPC. Ключ. Идентификатор. Структура статуса межпроцессового взаимодействия. Синхронизация на основе семафоров. Наборы семафоров. Создание семафора. Инициализация семафора. Операции с семафорами. Очереди сообщений. Создание очереди. Структуры данных, используемые для организации очередей сообщений. Помещение и выборка сообщений из очереди. Разделяемая память. Создание сегмента разделяемой памяти. Подключение сегмента к виртуальной памяти процесса.

		<p>Взаимодействие процессов посредством механизма IPC. Страничная организация памяти. Сегментная организация памяти. Сегментно-страничная организация памяти. Листание по требованию (demand paging). Алгоритмы подкачки. Локальные и глобальные алгоритмы. Пробуксовка (thrashing). Оптимальный алгоритм Биледи. Принцип локальности ссылок. Понятие рабочего набора. Откачка страниц. Процесс-stealer. Своппинг. Процесс-swapper. Удаленный вызов процедур RPC. Категории продуктов Middleware. Неявное сетевое программирование. Технология использования RPC. Схема функционирования RPC. Управление потоками в ОС Windows. Потоки в ОС Windows. Контекст потока. Состояние потока. Диспетчеризация и планирование потоков. Создание и завершение потока. Приостановка и возобновление потока. Управление процессами в ОС Windows. Процессы в ОС Windows. Создание и завершение процесса. Наследование дескрипторов. Дублирование дескрипторов. Обслуживание потоков. Управление файлами. Системные операции при работе с файловой системой. Алгоритм Open. Алгоритм Creat. Чтение и запись в файл. Алгоритм Read. Движение по файлу. Алгоритм Close. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при выполнении системных операций над файлами. Системные вызовы для работы с каталогами. Дополнительные системные вызовы при работе с файловой системой. Элементы файловой системы. Структура файловой системы. Структура индексного дескриптора. Суперблок. Выделение и освобождение дисковых блоков. Назначение описателя файла новому файлу. Таблица описателей файлов. Таблица файлов. Таблица открытых файлов.</p>		
ПК.10.В/ЭУ	35. Знать основные подходы в области системного моделирования	<p>Ресурсы ОС Unix. Функции подсистемы управления памятью. Функции подсистемы управления процессами. Функции подсистемы управления устройствами. Структура процесса. Таблица процессов. Промежуточная таблица процессов. Таблица областей. Контекст процесса. Информационное взаимодействие между процессами. Структура элемента таблицы процессов. U-область. Состояния процессов. Диаграмма состояний. Каналы. Наследование пользовательских дескрипторов. Понятие канала. Виды каналов. Средства организации непоименованных каналов. Алгоритм pipe. Создание поименованного канала. Чтение и запись через поименованный и непоименованный каналы. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при работе с каналом. Наследование пользовательских дескрипторов. Системный вызов Dup(). Системный вызов Fcntl(). Переназначение стандартного ввода и стандартного вывода. Клиент-серверные взаимодействия. Сервер. Клиент. Архитектура клиент-сервер. Сервисы. IP-адрес. Понятие серверного порта. Модель соединений. Модель дейтаграмм. Обобщенные сокеты. Сокеты для связи через сеть. Адресация в сокете. Дескриптор сокета. Создание сокета. Командный процессор Shell. Использование программного канала для передачи данных между процессами. Функциональная схема построения конвейера. Основной цикл программы Shell. Системные операции, связанные со временем. Написание программы на языке, моделирующей работу Shell-интерпретатора ОС Unix. Написание программы на языке Си для осуществления клиент-серверные взаимодействия посредством сокетов в режиме TCP-соединения в ОС Unix. Написание программы на языке Си для, осуществляющих межпроцессное взаимодействие основе семафоров, очередей</p>	Отчет по лабораторной работе (лабораторные работы 1-8) , вопросы к лабораторным работам. Отчет по РГЗ	Экзамен, вопросы 1-40. Вопросы тестов №1-6

сообщений и разделяемой памяти в ОС Unix. Написание программы на языке Си для синхронизации процессов посредством сигналов и каналов ОС Unix. Написание программы на языке Си для управления процессами ОС Unix. Написание программы на языке Си для управления ресурсами файловой системы ОС Unix. Написание программы на языке Си, моделирующих состояние системных таблиц ядра системы ввода-вывода ОС Unix. Обмен данными между параллельными процессами в ОС Windows. Способы передачи данных между процессами. Синхронный и асинхронный обмен данными. Работа с неименованными (анонимными) каналами. Работа с именованными каналами. Основные понятия ОС Unix. Процесс. Программа. Ядро. Функции ядра. Режим ядра и режим пользователя. Утилиты. Критическая секция. Системные вызовы. Схема обработки системных вызовов. Прерывания и их приоритеты. Мультипрограммирование и своппинг. Диспетчерский процесс. Общие аспекты управления памятью. Виртуальная и физическая память (1, 2, 7, 8). Основы теории семафоров. Понятие критической секции. Двоичный семафор. Общий семафор. Примитивы для работы с семафорами. P-операция. V-операция. Примеры использования P-, V-операций в задаче взаимного исключения. Использование семафоров в алгоритме банкира. Программирование в режиме TCP-соединения. Структура серверного процесса. Структура клиентского процесса (socket). Связывание (bind). Включение приема TCP-соединений (listen). Прием запроса на установку TCP-соединений (accept). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Программирование в режиме дейтаграмм. Структура серверного процесса. Структура клиентского процесса (socket). Связывание сервера с сокетом (bind). Связывание клиента с сокетом (bind). Подключение клиента (connect). Пересылка данных (send, recv). Закрытие TCP-соединения (close). Синхронизация процессов и потоков в ОС Windows. Критические секции. Примитивы синхронизации. События. Семафоры. Программная и аппаратная реализация синхронизации. Тупики. Обнаружение и предотвращение тупиков. Синхронизация процессов. Приостановление выполнения процесса. Алгоритм wait. Вызов программы в рамках порожденного процесса. Алгоритм exec. Сигналы. Взаимодействие процессов посредством сигналов. Моменты проверки и обработки сигналов на диаграмме состояний. Системный вызов Kill(). Алгоритм обработки сигналов. Системный вызов Signal(). Возможность побочных эффектов. Создание и завершение процесса. Понятие процесса. Идентификатор процесса. Наследование контекста среды при создании процесса. Системный вызов fork(). Группа процессов. Алгоритм fork. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при создании процесса. Способы завершения процесса. Алгоритм exit. Разделение родительским и порожденным процессами доступа к файлу. Взаимодействие процессов посредством каналов. Разделение процессами канала. Средства IPC в ОС Unix. Общие понятия средств IPC. Ключ. Идентификатор. Структура статуса межпроцессового взаимодействия. Синхронизация на основе семафоров. Наборы семафоров. Создание семафора. Инициализация семафора. Операции с семафорами. Очереди сообщений. Создание очереди. Структуры данных, используемые для организации очередей сообщений. Помещение и выборка сообщений из очереди. Разделяемая память. Создание сегмента разделяемой памяти.

		<p>Подключение сегмента к виртуальной памяти процесса. Взаимодействие процессов посредством механизма IPC. Удаленный вызов процедур RPC. Категории продуктов Middleware. Неявное сетевое программирование. Технология использования RPC. Схема функционирования RPC. Управление потоками в ОС Windows. Потоки в ОС Windows. Контекст потока. Состояние потока. Диспетчеризация и планирование потоков. Создание и завершение потока. Приостановка и возобновление потока. Управление процессами в ОС Windows. Процессы в ОС Windows. Создание и завершение процесса. Наследование дескрипторов. Дублирование дескрипторов. Обслуживание потоков. Управление файлами. Системные операции при работе с файловой системой. Алгоритм Open. Алгоритм Creat. Чтение и запись в файл. Алгоритм Read. Движение по файлу. Алгоритм Close. Модификация элементов таблиц файлов, описателей файлов, открытых файлов при выполнении системных операций над файлами. Системные вызовы для работы с каталогами. Дополнительные системные вызовы при работе с файловой системой. Элементы файловой системы. Структура файловой системы. Структура индексного дескриптора. Суперблок. Выделение и освобождение дисковых блоков. Назначение описателя файла новому файлу. Таблица описателей файлов. Таблица файлов. Таблица открытых файлов.</p>		
--	--	--	--	--

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 6 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.8, ПК.10.В/ЭУ. Время подготовки к экзамену – 2 дня.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 6 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (РГЗ). Требования к выполнению РГЗ, состав и правила оценки сформулированы в паспорте РГЗ.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.8, ПК.10.В/ЭУ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство

предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Паспорт экзамена

по дисциплине «Управление ресурсами в вычислительных системах», 6 семестр

1. Методика оценки

Экзамен проводится в частично устной, частично в письменной форме по билетам и тестам.

Экзамен включает две составляющие:

- тестирование по материалам лекций (максимум 24 балла);
- ответы на вопросы по экзаменационному билету (максимум 16 баллов).

При тестировании по материалам лекций проверяется знание студентом понятий по основным разделам курса. Примеры тестов приведены ниже.

Экзаменационный билет формируется по следующему правилу: первый вопрос – задача на написание программы, проверяющей знания материала по теории семафоров и средств межпроцессного взаимодействия, второй вопрос – проверка знаний основных понятий курса. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Примеры вопросов теста

Вопросы

5. Обслуживанием запросов пользователей занимается

- a) **утилиты**
- b) пользовательский процесс
- c) ядро
- d) технические средства

6. Интерпретатор команд shell

- a) **является процессом, выполняющимся в режиме задачи**
- b) является процессом, выполняющимся в режиме ядра
- c) для выполнения любой команды создает новый процесс
- d) осуществляет ввод командной строки, не пользуясь услугами ядра
- e) не пользуется системными вызовами

7. Своппинг - это

- a) управление внешней и оперативной памятью
- b) **перемещение процессов из оперативной памяти на диск и ввод их по мере необходимости обратно**
- c) управление процессами в оперативной памяти
- d) управление процессами во внешней памяти

8. Приоритет процесса является

- a) функцией от времени нахождения в пользовательской фазе
- b) **функцией от времени с момента последней загрузки в ОП**
- c) **функцией от времени использования процессора**
- d) функцией от времени нахождения в системной фазе

Номер вопроса	Правильные ответы
5	a
6	a
7	b
8	b, c

Вопросы

8. Интерпретатор команд выполняет командную строку:
`grep -w 'seven' f1.txt | sort | wc > f2.txt`,
 порождая три дочерних процесса и корректно организуя между ними непоименованный канал. Каково число записей в таблице открытых файлов?

- | | |
|-------|------|
| a) 13 | e) 9 |
| b) 12 | f) 8 |
| c) 11 | g) 7 |
| d) 10 | h) 6 |

9. Интерпретатор команд выполняет командную строку:
`grep -w 'seven' f1.txt | sort | wc > f2.txt`,
 порождая три дочерних процесса и корректно организуя между ними непоименованный канал. Каково число записей в таблице файлов?

- | | |
|------|------|
| a) 6 | d) 3 |
| b) 5 | e) 2 |
| c) 4 | f) 1 |

10. Интерпретатор команд выполняет командную строку:
`grep -w 'seven' f1.txt | sort | wc > f2.txt`,
 порождая три дочерних процесса и корректно организуя между ними непоименованный канал. Каково максимальное число ссылок в записях таблицы описателей файлов?

- | | |
|------|------|
| a) 5 | d) 2 |
| b) 4 | e) 1 |
| c) 3 | |

Номер вопроса	Правильные ответы
8	a
9	a
10	d

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ	Экзаменационный билет № 1
	По дисциплине <u>Управление ресурсами в выч. системах</u> Факультет ФПМИ курс <u>3</u> семестр <u>6</u>
Проверяется знание основных тем дисциплины:	
1. Два дочерних процесса выполняют некоторые циклы работ, передавая после окончания очередного цикла через один и тот же сегмент разделяемой памяти родительскому процессу очередные четыре строки некоторого стихотворения, при этом первый процесс передает нечетные четырехстишья, второй - четные. Циклы работ процессов не сбалансированы по времени. Родительский процесс компонует из передаваемых фрагментов законченное стихотворение и выводит его по завершении работы обоих процессов. Решить задачу с использованием аппарата семафоров	
2. Управление ресурсами в Unix. Основные понятия ОС (процесс, уровень ядра, уровень пользователя, утилиты, критическая секция, аппарат событий).	
Кафедра ТПИ	Преподаватель: _____

2. Критерии оценки

- Экзамен считается выполненным **неудовлетворительно**, если студент не написал программу из экзаменационного билета, слабо отвечал на вопросы тестов или экзаменационного билета и при этом набрал суммарно менее 10 баллов.
- Экзамен считается выполненным на **пороговом** уровне, если студент правильно написал программу из экзаменационного билета (возможно, с недочетами), но при этом слабо отвечал на вопросы теста или экзаменационного билета и при этом набрал более 10, но менее 20 баллов.
- Экзамен считается выполненным на **базовом** уровне, если студент правильно написал программу из экзаменационного билета, но при этом часто допускал ошибки при ответах на вопросы теста или экзаменационного билета и при этом набрал более 20, но менее 35 баллов.
- Экзамен считается выполненным на **продвинутом** уровне, если студент правильно написал программу из экзаменационного билета, дал правильные ответы на большинство вопросов теста или экзаменационного билета и при этом набрал более 35 баллов.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами аттестации балльно-рейтинговой системы:

Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки
98-100	A+	отлично
94-97	A	
90-93	A-	
87-89	B+	
84-86	B	хорошо
80-83	B-	
77-79	C+	
74-76	C	
70-73	C-	удовлетворительно
67-69	D+	удовлетворительно
64-66	D	
60-63	D-	
50-59	E	
25-49	FX	неудовлетворительно

Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки
0-24	F	

4. Вопросы к экзамену по дисциплине «Управление ресурсами в вычислительных системах»

1. Управление ресурсами в Unix. Основные понятия ОС (процесс, уровень ядра, уровень пользователя, утилиты, критическая секция, аппарат событий).
2. Мультипрограммирование и своппинг. Дисциплины мультипрограммирования.
3. Архитектура ОС Unix.
4. Управление процессами. Контекст процесса.
5. Состояние и взаимодействие процессов.
6. Выделение и освобождение дисковых блоков в ОС Unix.
7. Управление файлами.
8. Таблицы управление файлами. Системные операции при работе с файловой системой. Открытие и закрытие файлов.
9. Работа с каналами.
10. Наследование пользовательских дескрипторов.
11. Перенаправление ввода-вывода в Unix.
12. Управление процессами в ОС Unix. Порождение и завершение процессов
13. Синхронизация процессов.
14. Взаимодействие процессов посредством сигналов.
15. Основы теории семафоров.
16. Механизмы межпроцессного взаимодействия. Управление семафорами
17. Механизмы межпроцессного взаимодействия. Очереди сообщений
18. Механизмы межпроцессного взаимодействия. Сегменты разделяемой памяти.
19. Взаимодействие процессов в сети.
20. Программирование взаимодействия в режиме TCP-соединения.
21. Программирование взаимодействия в режиме дейтограмм.
22. Удаленный вызов процедур.
23. Управление ресурсами в ОС Windows. Основные понятия.
24. Планирование и диспетчеризация процессов.
25. Управление потоками в ОС Windows.
26. Создание, завершение потока.
27. Разработка оконных приложений.
28. Управление процессами в ОС Windows.
29. Создание, завершение процесса.
30. Синхронизация процессов и потоков в ОС Windows.
31. Критические секции.
32. Мьютексы.
33. События.
34. Семафоры.
35. Классы приоритетов процессов и приоритеты потоков.
36. Динамически подключаемые библиотеки.
37. Передача данных между процессами.
38. Работа с анонимными и именованными каналами в Windows.
39. Перенаправление ввода-вывода в Windows.
40. Почтовые ящики.

Паспорт расчетно-графического задания

по дисциплине «Управление ресурсами в вычислительных системах», 6 семестр

1. Методика оценки

Расчетно-графическое задание (РГЗ) по курсу «Управление ресурсами в вычислительных системах» ставит задачей практическое освоение вопросов управления ресурсами операционных систем (далее ОС) семейства Windows. Целью РГЗ является изучение основных особенностей ОС Windows: работа с потоками, организация графического интерфейса, механизма обработки сообщений, использование динамически подключаемых библиотек (dll), низкоуровневое взаимодействие с процессором.

Расчетно-графическое задание включает 4 уровня сложности. Студент самостоятельно принимает решение, в рамках какого уровня он будет выполнять индивидуальную работу.

В рамках первого уровня необходимо реализовать консольное Windows-приложение, в котором создаётся один дочерний поток, реализующий требования варианта задания и выводящий результат на консоль.

В рамках второго уровня необходимо реализовать минимальное графическое Windows-приложение, в котором при создании окна, но до момента его отображения на экране, создаётся дочерний поток, который реализует задачу в соответствии с выбранным вариантом, после чего полученный результат отображается в графическом окне.

При выполнении задания на третьем уровне необходимо реализовать все требований второго уровня с той разницей, что функция, выполняющая задание варианта задания, должна быть реализована в виде динамической библиотеки.

Четвёртый уровень предполагает полную реализацию требований третьего уровня, но дополнительно в функции из динамической библиотеки используется ассемблерная вставка, содержание которой зависит от номера варианта задания, который выполняется. Полученная информация из ассемблерной вставки выводится вместе с результатом решения задачи в соответствии с вариантом задания.

В представленных методических указаниях описываются программные средства, необходимые для реализации каждого уровня сложности. Для четвертого уровня сложности для некоторого варианта, аналогичного перечисленным ниже вариантам заданий, приведен полный текст основной программы и динамической библиотеки с описанием всех технологий, используемых в индивидуальной работе.

Для проверки выполненного расчетно-графического задания студент по электронной почте отправляет преподавателю, осуществляющему проверку выполнения РГЗ, на адрес его электронной почты письмо, содержащее комплект файлов с результатами выполнения индивидуальной работы. Требования к оформлению письма и составу файлов содержатся в методических указаниях по выполнению РГЗ. Кроме этого студент предоставляет в печатном виде отчет о выполнении РГЗ.

Обязательные структурные части РГЗ:

- 1) Описание набора программных средств, используемых в программе.
- 2) Описание логики и алгоритма программы.
- 3) Листинг программы.
- 4) Набор необходимых тестов.

5) Анализ работы программы и выполненных тестов.

При оценке выполнения исследовательской работы учитываются:

- правильность написания программы для указанного варианта задания;
- понимание логики программы, а также теоретического материала, на основании которого эта программа написана;
- уровень документированности программы;
- достаточность и качество предлагаемых тестов;
- качество оформления отчета по РГЗ.

Результаты проверки РГЗ преподаватель выставляет на своей странице на портале НГТУ.

2. Критерии оценки

- Работа считается **не выполненной**, если написанная студентом программа не выполняет или выполняет некорректно действия, предусмотренные вариантом задания и выбранным уровнем сложности, либо уровень оригинальности РГЗ составляет менее 20%; оценка составляет 0 баллов.
- Работа считается выполненным на **пороговом** уровне, если написанная студентом программа выполняет действия, предусмотренные вариантом задания и выбранным уровнем сложности, уровень оригинальности РГЗ составляет не менее 40%, но при этом содержатся значительные недоработки в проверяемых элементах РГЗ; оценка составляет минимум баллов для выбранного уровня сложности (I уровень – 4, II уровень – 6, III уровень – 8, IV уровень – 10 баллов).
- Работа считается выполненным на **базовом** уровне, если написанная студентом программа выполняет действия, предусмотренные вариантом задания и выбранным уровнем сложности, уровень оригинальности РГЗ составляет не менее 65 %, но при этом содержатся незначительные недоработки в проверяемых элементах РГЗ; оценка определяется диапазоном баллов для выбранного уровня сложности (I уровень – 5-6, II уровень – 7-10, III уровень – 9-14, IV уровень – 11-17 баллов).
- Работа считается выполненным на **продвинутом** уровне, если написанная студентом программа выполняет действия, предусмотренные вариантом задания и выбранным уровнем сложности, уровень оригинальности РГЗ составляет не менее 75 % и все проверяемые элементы РГЗ тщательно описаны; оценка составляет максимум баллов для выбранного уровня сложности (I уровень – 7, II уровень – 11, III уровень – 15, IV уровень – 18 баллов).

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами аттестации балльно-рейтинговой системы:

Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки
98-100	A+	отлично
94-97	A	
90-93	A-	
87-89	B+	

Диапазон баллов рейтинга	Оценка ECTS	Традиционная (4-уровневая) шкала оценки
84-86	B	хорошо
80-83	B-	
77-79	C+	
74-76	C	
70-73	C-	удовлетворительно
67-69	D+	удовлетворительно
64-66	D	
60-63	D-	
50-59	E	
25-49	FX	неудовлетворительно
0-24	F	

4. Примерный перечень тем РГЗ

№	Задание для выполнения I, II, III уровня сложности	Дополнительное задание для выполнения IV уровня сложности
1	Определить приоритет текущего потока	Определить наличие сопроцессора
2	Определить приоритет текущего процесса	Определить наличие поддержки виртуального режима работы процессора
3	Определить количество мониторов	Определить наличие поддержки команды RDTSC
4	Определить количество кнопок мыши	Определить наличие команды сравнения и обмена 8 битного значения
5	Определить максимальную ширину окна (в пикселях)	Определить наличие поддержки MMX технологии
6	Определить максимальную высоту окна (в пикселях)	Определить наличие поддержки технологии SSE
7	Определить ширину экрана (в пикселях)	Определить наличие поддержки технологии SSE2
8	Определить высоту экрана (в пикселях)	Определить наличие поддержки технологии SSE3
9	Определить наличие сетевого подключения	Определить наличие поддержки технологии Hyper Threading
10	Определить число процессоров	Определить размер страницы буфера

	(ядер)	ассоциативной трансляции команд*
11	Определить размер страницы памяти	Определить ассоциативность и количество входов буфера ассоциативной трансляции команд*
12	Определить тип процессоров (ядер)	Определить размер страницы буфера ассоциативной трансляции данных*
13	Определить текущую директорию	Определить ассоциативность и количество входов буфера ассоциативной трансляции данных*
14	Определить имя компьютера	Определить размер КЭШа данных первого уровня*
15	Определить имя пользователя	Определить ассоциативность КЭШа данных первого уровня*
16	Определить системную директорию	Определить длину строки КЭШа данных первого уровня*
17	Определить директорию Windows	Определить размер КЭШа команд первого уровня*
18	Определить текущий год	Определить ассоциативность КЭШа команд первого уровня*
19	Определить текущий день	Определить длину строки КЭШа команд первого уровня*
20	Определить текущий день недели	Определить размер КЭШа данных второго уровня*
21	Определить текущий месяц	Определить ассоциативность КЭШа данных второго уровня*
22	Определить текущий час	Определить длину строки КЭШа данных второго уровня*
23	Определить текущую минуту	Определить размер КЭШа данных третьего уровня*
24	Определить тип клавиатуры	Определить ассоциативность КЭШа данных третьего уровня*
25	Определить количество функциональных клавиш клавиатуры	Определить длину строки КЭШа данных третьего уровня*