

Операционные системы

Дмитрий Валентинович Иртегов

Et.nsu.ru, дата размещения 04.12.2014

1. Аннотация

Дисциплина «Операционные системы» имеет своей целью: изучение студентами архитектуры современных операционных систем. Студенты получают представление о сервисах, предоставляемых современными операционными системами (ОС) и о приемах реализации этих сервисов, ознакомятся с обзором реальных архитектур сложных, исторически сложившихся программных комплексов. На практических занятиях студенты ознакомятся с системным интерфейсом ОС семейства Unix (Sun Solaris).

2. Место дисциплины в структуре образовательной программы

Дисциплина «Операционные системы» относится к базовой (общепрофессиональной) части профессионального цикла подготовки по направлению 230100 «Информатика и вычислительная техника».

Дисциплина «Операционные системы» опирается на следующие дисциплины данной ООП:

- Информатика;
- Математическая логика и теория алгоритмов;
- Дискретная математика;
- Теория вероятностей и математическая статистика;
- Программирование;
- Основы объектно-ориентированного программирования;
- ЭВМ и периферийные устройства;
- Сетевые технологии.

Результаты освоения дисциплины «Операционные системы» используются в следующих дисциплинах данной ООП:

- Объектно-ориентированный анализ и дизайн;

- Основы параллельного программирования;
- Сети и телекоммуникации;
- Управление производственным процессом разработки программного обеспечения;
- Защита информации.

Для успешного усвоения материала, обучающийся должен владеть языками программирования С и Java и техникой программирования, навыками решения основных задач программирования, техникой отдельной компиляции, отладчиками, знать классические структуры данных и алгоритмы и уметь оценивать их вычислительную сложность, знать основы теории вероятностей и математической статистики, уметь применять математические методы для решения практических задач, в частности, уметь вычислять вероятности различного сорта событий, понимать основные принципы организации и функционирования ЭВМ фон-неймановской архитектуры, уметь сопоставлять внутреннее (машинный код) и текстовое (ассемблер) представления исполнительного кода и исходный код программы (ЯВУ), знать TCP socket API.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Дисциплина нацелена на формирование у обучающегося:

профессиональных компетенций (ПК):

ПК-2	осваивать методики использования программных средств для решения практических задач
ПК-5	разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования
ПК-17	Знает принципы построения современных операционных систем и особенности их применения
ПК-18	Владеет навыками решения типовых задач системного программирования современных ОС
ПК-35	Понимает основные принципы организации и функционирования ЭВМ
ПК-41	Понимает основные принципы организации и функционирования компьютеров с общей и распределенной памятью

инструментальных компетенций (ИК):

ИК-1	Имеет навыки использования инструментальных средств программирования для ОС семейства Unix
------	--

Также дисциплина участвует в формировании у обучающегося:

общекультурных компетенций (ОК):

ОК-11	осознает сущность и значение информации в развитии современного общества; владеет основными методами, способами и средствами получения, хранения, переработки информации
ОК-12	имеет навыки работы с компьютером как средством управления информацией

профессиональных компетенций (ПК):

ПК-6	обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности ;
ПК-9	участвовать в настройке и наладке программно-аппаратных комплексов
ПК-10	сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем
ПК-11	инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем
ПК-42	Владеет основными методами и средствами разработки параллельных программ для компьютеров с общей и распределенной памятью
ПК-43	Владеет методами эффективного программирования для компьютеров с общей и распределенной памятью
ПК-47	Имеет представление о корректности параллельных программ и способах её достижения
ПК-51	Имеет представление о корректности параллельных программ и способах её достижения
ПК-62	Способен формировать комплекс мер по информационной безопасности с учетом его правовой обоснованности, административно-управленческой и технической реализуемости и экономической целесообразности
ПК-63	Способен организовывать и поддерживать выполнение комплекса мер по информационной безопасности, управлять процессом их реализации с учетом решаемых задач и организационной структуры объекта защиты, внешних воздействий, вероятных угроз и уровня развития технологий защиты информации
ПК-69	Знает основы организации параллельной/распределенной обработки информации

инструментальных компетенций (ИК):

ИК-4	владеет языком программирования высокого уровня Си и техникой программирования
ИК-5	имеет навыки решения типовых задач программирования и владеет основными моделями и методами их решения
ИК-6	владеет техникой раздельной компиляции, отладчиком и другими возможностями интегрированных сред разработки

В результате освоения дисциплины обучающийся должен:

Знать:

- основы системного программирования;
- принципы построения современных операционных систем и особенности их применения;
- методы и средства обеспечения информационной безопасности компьютерных систем;
- основные принципы функционирования современных ОС и их подсистем: загрузчика, системы управления виртуальной памятью, планировщика, подсистемы ввода-вывода, файловой системы, подсистем аутентификации и авторизации.

Уметь:

- выбирать, комплексировать и эксплуатировать программно-аппаратные средства в создаваемых вычислительных и информационных системах и сетевых структурах;
- устанавливать, тестировать, испытывать и использовать программно-аппаратные средства вычислительных и информационных систем;
- решать типовые задачи системного программирования в современных ОС;
- пользоваться документацией, в том числе встроенной документацией man в системах семейства Unix;

Владеть:

- навыками работы с различными операционными системами и их администрирования;

- инструментальными средствами программирования для ОС семейства Unix: командными процессорами sh/bash, компиляторами командной строки (GCC, Oracle Solaris Studio), утилитой make, отладчиками gdb или dbx, интегрированными средами Eclipse или Netbeans/Oracle Solaris Studio.

4. Структура и содержание дисциплины «Операционные системы»

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Лабораторные работы	Самостоятельная работа	Экзамен	
1	Классификация операционных систем	4	1	2				
2	Среда исполнения программ в Unix. Файловый ввод-вывод	4	1-4	4	8	10		
2.1	Лекции		2-3	4				
2.2	Лабораторные работы		1-4		8	10		Сдача задач
3	Сборка и загрузка программ	4	4	2				
4	Создание процессов и исполнение программ в Unix	4	5-7	4	6	8		
4.1	Лекции		5-6	4				
4.2	Лабораторные работы		5-7		6	8		Сдача задач
5	Виртуальная память	4	7	2				
6	Управление файлами и каталогами в Unix	4	8-11	4	8	10		
6.1	Лекции		8-9	4				
6.2	Лабораторные работы		8-11		8	10		Сдача задач
7	Многопоточное исполнение и синхронизация потоков	4	10	2				
8	Межпроцессное взаимодействие в Unix	4	11-16	10	10	14		
8.1	Лекции		11-15	10				
8.2	Лабораторные работы		12-16		10	14		Сдача задач
9	Реализация многопоточности	4	16	2				
10	Промежуточная аттестация	4	17			2		Дифференцированный зачет
	ИТОГО по семестру	4	1-17	32	32	44		

11	Драйверы внешних устройств	5	1-3	6				
12	Многопоточное исполнение в Unix (POSIX - Threads)	5	1-6	6	12	3		
12.1	Лекции		4-6	6				
12.2	Лабораторные работы		1-6		12	3		Сдача задач
13	Файловые системы	5	7-9	6				
14	Синхронизация потоков в Unix	5	7-12	6	12	3		
14.1	Лекции		10-12	6				
14.2	Лабораторные работы		7-12		12	3		Сдача задач
15	Обработка ошибок и исключений	5	13	2				
16	Вопросы безопасности	5	14-15	4				
17	Асинхронный и событийно-ориентированный ввод/вывод	5	13-16	2	8	2		
17.1	Лекции		16	2				
17.2	Лабораторные работы		13-16		8	2		Сдача задач
18	Промежуточная аттестация	5	19-22				36	Экзамен
	ИТОГО по семестру	5	1-22	32	32	8	36	
	ИТОГО по курсу			64	64	52	36	

5. Образовательные технологии

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента.

Основными образовательными технологиями, используемыми при изучении дисциплины, являются:

- *Проблемное обучение:*

При выполнении задач студент самостоятельно проводит поиск по документации. Приветствуется использование приемов программирования и системных вызовов, не изучаемых в лекционном материале.

- *Контекстное обучение:*

При решении задач, студент должен сопоставить теоретические знания о принципах организации виртуальной памяти, многопоточности, ввода-вывода и др. с практическим применением этих принципов при программировании в среде Unix.

- *Обучение на основе опыта:*

В теоретической части курса, предполагается, что студенты интегрируют полученную теоретическую информацию с практическими навыками, получаемыми в ходе лабораторных работ и при повседневной эксплуатации личных компьютеров, смартфонов и других вычислительных устройств.

- *Междисциплинарное обучение:*

Материал курса существенным образом опирается на материалы курсов «ЭВМ и периферийные устройства» и «Программирование», и тесно взаимосвязана с материалами параллельно изучаемых курсов «Основы параллельного программирования», «Сетевые технологии», «Базы данных».

- *Опережающая самостоятельная работа:*

В ходе лабораторных работ приветствуется опережающая по сравнению с лекционным материалом сдача задач.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В ходе самостоятельной работы студенты выполняют практические работы, а в ходе лабораторных работ — сдают эти работы преподавателю. Выполнение указанных видов работ является обязательным для всех студентов, а результаты текущего контроля служат основанием для выставления оценок в ведомость контрольной недели на факультете.

Оценка за контрольную неделю выставляется в зависимости от количества сданных задач:

1 контрольная неделя:

менее 2 задач - «неудовлетворительно»

от 2 до 3 задач - «удовлетворительно»

от 4 до 6 задач - «хорошо»

более 6 задач - «отлично»

2 контрольная неделя:

менее 4 задач - «неудовлетворительно»

от 4 до 7 задач - «удовлетворительно»

от 8 до 14 задач - «хорошо»

более 14 задач - «отлично»

Промежуточная аттестация по дисциплине за 4 семестр проводится в форме дифференцированного зачета. Оценка за диф.зачет выставляется на основе количества сданных задач:

менее 6 задач - «неудовлетворительно»

от 6 до 10 задач - «удовлетворительно»

от 11 до 22 задач - «хорошо»

23 задачи - «отлично».

Промежуточная аттестация по дисциплине за 5 семестр проводится в форме сдачи устного экзамена. Каждый экзаменационный билет включает в себя два теоретических вопроса. Оценка за практические работы в течение семестра учитывается на экзамене. Оценка за практические работы выставляется на основе количества сданных задач:

менее 6 задач - «неудовлетворительно»

от 6 до 10 задач - «удовлетворительно»

от 11 до 22 задач - «хорошо»

23 задачи - «отлично».

При оценке за практические работы «удовлетворительно», студент имеет право получить на экзамене итоговую оценку не выше, чем «хорошо». При оценке за практические работы «хорошо», студент имеет право получить на экзамене итоговую оценку вплоть до «отлично». При оценке за практические работы «отлично», студент получает итоговую оценку «отлично» без опроса на экзамене.

6.1. Задачи для самостоятельной и практической работы 4 семестра:

<http://ccfit.nsu.ru/~deviv/courses/unix/tasks.html>

6.2. Задачи для самостоятельной и практической работы 5 семестра:

[http://swsoft.nsu.ru/WackoWiki/KursOperacionnyeSistemy/PraktikumPosixThreads/PtheadTasks](http://swsoft.nsu.ru/WackoWiki/KursOperacionnyeSistemy/PraktikumPosixThreads/PthreadTasks)

6.3 Список вопросов к экзамену:

1. Алгоритм работы библиотечных функций malloc/free GNU LibC.
2. Алгоритмы поиска жертвы при страничном обмене и кэшировании. Критерии выбора и влияние алгоритма на производительность. Что такое рабочее множество страниц?
3. Аутентификация и проверка подлинности кода в Lotus Notes.
4. Ввод-вывод в режиме опроса и по прерываниям. Преимущества и недостатки.
5. Динамическое выделение памяти. Методы борьбы с фрагментацией. Основные алгоритмы выделения памяти.
6. Динамическое выделение памяти в ОС семейства Unix и стандарте POSIX.
7. Диспетчер задач в транспьютере.
8. Драйвер устройства. Функции драйвера в ОС семейства Unix.
9. Журнальные файловые системы. Принципы работы. Для чего это нужно?
10. Загружаемые модули и разделяемые библиотеки Win32 (PE).
11. Загрузка ОС на PC-совместимых компьютерах.
12. Запуск задач в ОС семейства Unix.
13. Инверсия приоритета. Способы ее предотвращения и способы обхода этой проблемы.
14. Как происходит загрузка операционной системы? Что такое первичный загрузчик? Вторичный? Как происходит загрузка бездисковых машин?
15. Как реализуется многопоточность на однопроцессорной машине. Что такое контекст процесса? Какие особенности процессора влияют на скорость переключения процессов?
16. Кооперативная и вытесняющая (preemptive) многозадачность. Преимущества и недостатки обеих архитектур.
17. Линки в транспьютере.

18. Мертвая и живая блокировки. Способы их предотвращения. Преимущества и недостатки каждого из методов.
19. Методы реализации виртуальной памяти. Базовая адресация, сегментная и страничная виртуальная память.
20. Механизм setuid в ОС семейства Unix.
21. Объектный модуль. Объектная библиотека. Структуры данных, содержащиеся в объектном модуле, в общих чертах. Алгоритм работы сборщика и выбора модулей из библиотеки.
22. Определение задачи реального времени. Чем системы РВ отличаются от систем разделенного времени? Пример архитектуры ОС реального времени.
23. Организация страничного обмена в VMS, OpenVMS и Windows NT.
24. Организация файловой системы HPFS.
25. Планировщики разделенного времени. Динамическое управление приоритетами в системах разделенного времени.
26. Подсистема виртуальной памяти и алгоритм поиска жертвы в Windows NT.
27. Понятия инода и связи в файловых системах ОС семейства Unix.
28. Почтовые ящики (mailbox) в VAX/VMS.
29. Права доступа к файлам в ОС семейства Unix.
30. Прерывания в классических процессорах (PDP-11, 8086, x86). Внешние прерывания и исключения (exceptions).
31. Приоритеты процессов в OS-9.
32. Приоритеты процессов и нитей. Управление приоритетами для нитей реального и разделенного времени. Где используется и для чего нужно динамическое изменение приоритета?
33. Программные каналы (трубы) в системах семейства Unix.
34. Разделяемая память. Преимущества и недостатки по сравнению с другими методами межпроцессного взаимодействия.
35. Разделяемые библиотеки формата ELF.
36. Распределение памяти алгоритмами близнецов и парных меток Ограничения этих алгоритмов.
37. Реентерабельная программа. Техника реализации реентерабельных программ.

- Всегда ли это возможно? Что такое критическая секция?
38. Сборка в момент загрузки. Преимущества и недостатки этого метода. Чем отличаются DLL Win32 и разделяемые библиотеки ELF.
 39. Сборка мусора. Основные стратегии сборки мусора, их преимущества и недостатки.
 40. Семафоры Дийкстры. Мутексы, двоичные семафоры и семафоры общего вида. Мертвая блокировка и способы избежать ее.
 41. Семафоры Unix System V IPC. Наборы семафоров.
 42. Сигналы в системах семейства Unix.
 43. Системы управления доступом. Полномочия и списки контроля доступа. Кольца доступа.
 44. Событийно-ориентированные системы. Обязательно ли такая система является многопоточной?
 45. Спинлоки и их применение. Их преимущества и недостатки по сравнению с другими средствами взаимного исключения.
 46. Структура и особенности организации файловой системы UFS (FFS).
 47. Структура и принципы работы файловой системы NTFS.
 48. Структура файловой системы RT-11.
 49. Троянские программы и способы их внедрения. Меры по защите от троянских программ.
 50. Уровни RAID.
 51. Устойчивые к сбоям файловые системы. Методы реализации устойчивых ФС.
 52. Файловая система FAT.
 53. Файловая система ISO 9660 (CDFS).
 54. Файловая система NetApp WAFL.
 55. Флаги событий в RSX и VMS. Что такое AST?
 56. Формирование запросов на ввод/вывод в системах фирмы DEC (RSX-11, VMS, OpenVMS). Какие преимущества предоставляет этот метод?
 57. Что такое абсолютный и относительный загрузчики? Структура абсолютного и перемещаемого загрузочных модулей. Что такое позиционно-независимый код?

58. Что такое гармонически взаимодействующие последовательные процессы?
Средства для реализации этой дисциплины в существующих системах.

7. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Иртегов Д.В., Введение в операционные системы, 2-е изд. БХВ-Петербург, 2008, 1040 стр, ил.
2. Таненбаум, Эндрю С. Современные операционные системы = Modern Operating Systems : [пер. с англ.] / Э. Таненбаум .— 2-е изд. — СПб. и др. : ПИТЕР, 2007 .— 1037 с. : ил. ISBN 978-5-318-00299-1
3. Системные вызовы и библиотеки Unix SVR4, НГУ 2014, <http://lib.nsu.ru:8080/xmlui/handle/nsu/979>
4. Иртегов Д.В., Многопоточное программирование с использованием POSIX Thread Library, НГУ, 2008. Текст также доступен по адресу http://sun.nsu.ru/tranings/download.jsp?path=Programming/Irtegov/Irtegov_all.zip

б) дополнительная литература:

1. Баррон Д. Ассемблеры и загрузчики. — М.: Мир, 1974.
2. Вахалия Ю. Unix изнутри, - СПб: Питер, 2003
3. Дейкстра Э. Дисциплина программирования. — М.: Мир, 1978.
4. Дейтел Г. Введение в операционные системы. — М.: Мир, 1987.
5. Кнут Д. Э. Искусство программирования. — Издательский дом Вильямс, 2000.
6. Керниган Б., Пайк Р. Unix — универсальная среда программирования. — М.: Финансы и статистика, 1992.
7. Робачевский А. Операционная система Unix. — СПб.: БХВ — Санкт-Петербург, 2012.
8. Русинович М., Соломон Д. Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000, 4-е издание – СПб: Питер, 2006
9. Стивенс У., Феннер Б., Рудофф Э., Unix: разработка сетевых приложений. 3-е изд. - СПб: Питер, 2006
10. Стивенс У., Раго С., UNIX. Профессиональное программирование, СПб: Символ-Плюс, 2007
11. Хоар Ч. Взаимодействующие последовательные процессы. — М.: "Мир", 1989.

12. Хэвиленд К., Грэй Д., Салама Б. Системное программирование в UNIX. — М.: ДМК Пресс, 2000.
13. Mauro J., McDougal R., Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture (2nd Edition), Prentice Hal, 2006.

в) программное обеспечение и Интернет-ресурсы:

1. OpenSolaris или Solaris 10 Express
2. Встроенное системное руководство man
3. Oracle Solaris Studio
4. Сайт проекта OpenIndiana (содержит исходные тексты OpenSolaris):
<http://openindiana.org/>

8. Материально-техническое обеспечение дисциплины

Лекционная поточная аудитория:

- Ноутбук, медиа-проектор, экран.
- Программное обеспечение для демонстрации слайд-презентаций.

Терминальный класс:

- 10 рабочих станций или виртуальных машин на рабочих станциях с процессорами x86 (желательно x64) или SPARC v9, не менее 1Гб ОЗУ, не менее 10Гб дискового пространства, с видео- и сетевыми адаптерами и дисковыми контроллерами, поддерживаемыми ОС Solaris 10/11 или OpenIndiana
- Сервер, физическая или виртуальная машина с процессором x86 (желательно x64) или SPARC v9, желательно не менее 2 процессоров, не менее 2Гб ОЗУ, не менее 100Гб дискового пространства, с сетевым адаптером и дисковым контроллером, поддерживаемым ОС Solaris 10/11 или OpenIndiana.

Рецензент (ы) _____

Программа одобрена на заседании Методической комиссии ФИТ
от _____ года, протокол № _____.