

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе
и образовательным инновациям

О.И. Чуприс

(подпись)

20.06.2018

(дата утверждения)

Регистрационный № УД- 6081 /уч.

**АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

1-31 03 01 Математика (по направлениям)

Направление специальности

1-31 03 01-04 Математика (научно-конструкторская деятельность)

2018 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 03 01-2013 (30.08.2013) и учебного плана G31-209/уч. (29.05.2015) специальности 1-31 03 01 Математика (по направлениям), направление 1-31 03 01-04 Математика (научно-конструкторская деятельность).

Составители:

Гладков А. Л. – заведующий кафедрой математической кибернетики механико-математического факультета Белорусского государственного университета, доктор физико-математических наук, профессор;

Зайцев В. С., старший преподаватель кафедры математической кибернетики механико-математического факультета Белорусского государственного университета.

Рекомендована к утверждению:

кафедрой математической кибернетики механико-математического факультета Белорусского государственного университета (протокол № 9 от 29.04.2018);

учебно-методической комиссией механико-математического факультета Белорусского государственного университета (протокол № 8 от 19.06.2018).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Тенденция развития общества последних десятилетий во всем мире состоит в информатизации всех областей жизни. Создаются все новые и более современные программные и аппаратные решения, цель которых состоит в улучшении качества жизни, решении рутинных задач, как в повседневной жизни так и производстве.

Проектирование и производство таких систем требует должного уровня и профиля подготовки специалистов, которые будут конструировать и разрабатывать их. Сложность таких систем неуклонно растет с каждым годом, и проектирование новых требует знания и использования уже существующих технологий как базис.

Маршрут проектирования современной цифровой аппаратуры содержит множество этапов. Если говорить о проектировании цифровой аппаратуры, то это разработка математической модели будущего устройства, затем проектирование модели на языке описания аппаратуры, синтез схемы проектирование топологии, изготовление на производстве. Каждый шаг маршрута проектирования содержит множество под этапов и на каждом из них используется специализированные ПО. Специалист высокого уровня должен не только знать и владеть методикой использования этого ПО, но также уметь автоматизировать задачи возникающие на стыке их использования.

На стыке использования результатов, возникает отдельный круг задач, когда нужно получить выходные данные на одном из этапов и передать их на следующий этап или обработать эти данные. Для их решения существуют уже разработанные инструменты (специализированное ПО), которые являются универсальными и применяется для очень широкого круга задач автоматизации работы системы под управлением ПК.

Цель дисциплины: получение студентами профессиональных навыков в области автоматизации повторяющихся задач в процессе разработки ПО и проектирования аппаратуры. Обучающиеся должны освоить системные знания в области автоматизации запуска существующих САПР, анализа и документирования полученных в ходе проектирования результатов.

Развивающая цель: получение студентами практических навыков работы со скриптовым языком автоматизации запуска ПО (bash), специализированным ПО автоматизации запуска приложений (make, Jenkins), специализированным ПО позволяющим автоматизацию создания документации разработанного приложения или аппаратуры (doxygen).

Воспитательная цель: развитие у студентов навыка самостоятельного поиска и получения информации в литературе и сети интернет, который в будущем может упростить им решение возникающих задач в проектировании и в дальнейшей их повседневной жизни.

Задача дисциплины состоит в изучении методики использования решений позволяющих автоматизировать процесс запуска и обработки результатов существующего ПО.

В результате изучения дисциплины «Автоматизация проектирования программного обеспечения» студент должен:

знать: синтаксис и основные конструкции скриптового языка bash; основные консольные утилиты обработки текстовой информации; методы работы с ПО make, Jenkins; формат комментариев используемых для автоматической генерации документации по коду с помощью Doxygen;

уметь: разрабатывать скрипты на языке bash для запуска и обработки результатов консольных приложений; разрабатывать сценарии для ПО make; создавать задачи и графики запуска в системе Jenkins; генерировать документацию по коду с помощью Doxygen

Учебная дисциплина строится таким образом, чтобы обучающийся приобрел следующие компетенции специалиста:

Академические:

АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.

АК-2. Владеть системным и сравнительным анализом.

АК-3. Владеть исследовательскими навыками.

АК-4. Уметь работать самостоятельно.

АК-5. Быть способным вырабатывать новые идеи (обладать креативностью).

АК-6. Владеть междисциплинарным подходом при решении проблем.

АК-7. Иметь навыки, связанные с использованием технических устройств, управлением информацией и работой с компьютером.

АК-8. Обладать навыками устной и письменной коммуникации.

АК-9. Уметь учиться, повышать свою квалификацию в течение всей жизни.

Социально-личностные:

СЛК-2. Быть способным к социальному взаимодействию.

СЛК-3. Обладать способностью к межличностным коммуникациям.

СЛК-5. Быть способным к критике и самокритике.

СЛК-6. Уметь работать в команде.

Профессиональные:

ПК-1. Разрабатывать практические рекомендации по использованию научных исследований, планировать и проводить экспериментальные исследования, исследовать патентоспособность и показатели технического уровня разработок программного обеспечения информационных систем.

ПК-4. Разрабатывать и тестировать информационные системы, осуществлять защиту приложений и данных.

ПК-9. Осуществлять выбор оптимального варианта проведения научно-исследовательских работ.

ПК-13. Взаимодействовать со специалистами смежных профилей.

ПК-16. Готовить доклады, материалы к презентациям.

ПК-22. Осваивать и реализовывать управленческие инновации в сфере высоких технологий.

ПК-27. Реализовывать инновационные проекты в профессиональной деятельности.

Дисциплина «Автоматизация проектирования программного обеспечения» относится к циклу дисциплин специализации.

Материал данной дисциплины тесно связан с дисциплиной «Методы программирования и информатика».

Учебная программа по дисциплине «Автоматизация проектирования программного обеспечения» предназначена для студентов 2 курса (4 семестр) очной формы получения высшего образования.

На изучение учебной дисциплины по специальности 1-31 03 01 «Математика (по направлениям)» (1-31 03 01-01 Математика (научно-конструкторская деятельность)» предусмотрено 60 часов, в том числе 46 часов аудиторных занятий, включая 28 часов лекций, 14 часов лабораторных занятий, 4 часа УСР. Текущая аттестация – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Введение в задачи автоматизации на стыке этапов маршрута проектирования ПО и аппаратуры.

Проектирование цифровых систем на кристалле. Обзор используемых САПР.

Обзор существующего ПО и подходов к его использованию, позволяющих автоматизировать и тем самым ускорить процесс проектирования.

Тема 2. Введение в скриптовый язык bash

Заголовок скрипта и комментарии. Переменные. Позиционные параметры (аргументы). Проверка условий. Операции проверки файлов. Функции. Циклы. Операторы выбора. Операторы перенаправления вывода. Конвейеры. Команды для работы с файлами. Перехват сигналов. Методы отладки скрипта.

Тема 3. Основные приемы работы с консольным ПО обработки текстовой информации

Консольные утилиты: cat, cd, pwd, cp, mkdir, chmod, chown, head, tail, ls, grep, tee, bg, fg, setenv, df, du, alias, find, awk, sed, vim. Основные приемы работы.

Тема 4. Основные приемы работы с ПО make.

Переменные. Комментарии. Передача аргументов. Создание и вызов цели. Запуск цепочки целей. Встроенные в строку сравнения и циклы. Запуск сторонних утилит.

Тема 5. Jenkins как система автоматизации запуска ПО и контроля полученных результатов

Создание сценария для запуска. Задание расписания. Настройка триггеров постобработки. Сохранение и публикация результатов запуска.

Тема 6. ПО Doxygen для генерации документации из кода.

Заголовки. Списки. Автоматические ссылки. Иллюстрации. Таблицы. Настройки структуры документа. Задание цветовой схемы.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Введение в задачи автоматизации на этапах маршрута проектирования ПО и аппаратуры	2						Собеседование
1.1	Обзор маршрута проектирования цифровых блоков и использование скриптов для автоматизации этого маршрута	2						
2	Введение в скриптовый язык bash	8			4		2	Самостоятельная работа
2.1	Базовый синтаксис языка bash. Переменные, операторы, входные аргументы	4			2			
2.2	Функции, работа с файлами, конвейеризация. Перехват сигналов. Отладка скриптов.	4			2			
3	Основные приемы работы с консольным ПО обработки текстовой информации	10			4			Тест
3.1	Утилиты работы с файловой системой.	6			2			
3.2	Утилита обработки sed, регулярные выражения. Синтаксис при работе с awk.	4			2			
4	Основные приемы работы с ПО make.	4			2			Контрольная работа
4.1	Переменные. Комментарии. Передача аргументов. Создание и	4			2			

	вызов цели. Запуск цепочки целей. Встроенные в строку сравнения и циклы. Запуск сторонних утилит.							
5	Jenkins, как система автоматизации запуска ПО и контроля полученных результатов	2			2		2	Самостоятельная работа
5.1	Создание сценария для запуска. Задание расписания. Настройка триггеров постобработки. Сохранение и публикация результатов запуска.	2			2			
6	ПО Doxygen для генерации документации из кода.	2			2			Тест
6.1	Заголовки. Списки. Автоматические ссылки. Иллюстрации. Таблицы. Настройки структуры документа. Задание цветовой схемы.	2			2			
	ИТОГО	28			14		4	

ИНФОРМАЦИОННО - МЕТОДИЧЕСКАЯ ЧАСТЬ

Рекомендуемая литература

Основная:

1. Machtelt Garrels, "[Bash Guide for Beginners](#)", 9.02.2010 (Revision 1.12)
Перевод: Н.Ромоданов, Дата перевода: март 2011 г.
https://www.opennet.ru/docs/RUS/bash_scripting_guide/
2. Mendel Cooper, Advanced Bash-Scripting Guide, Искусство программирования на языке сценариев командной оболочки.
3. Bash FAQ, version 4.14, for Bash version 4.4.
<ftp://ftp.cwru.edu/pub/bash/FAQ>
4. Руководство по GNU make (версия 3.79)
http://www.linuxlib.ru/prog/make_379_manual.html
5. GNU Make Manual Free Software Foundation last updated May 22, 2016.
<http://www.gnu.org/software/make/manual/>
6. Managing Projects with GNU Make, Third Edition The Power of GNU Make for Building Anything By Robert Mecklenburg November 2004 Pages: 300 <http://www.oreilly.com/openbook/make3/book/index.csp>
7. Anatole Olczak, Bourne Shell Quick Reference Guide, ASP, Inc., 1991, 093573922X.

Дополнительная:

1. Документирование исходного кода для автоматизированного построения программной документации посредством пакета DoxyGen.
http://www.nrjetix.com/fileadmin/doc/publications/additional_info/DoxyGen%20guide.pdf
2. Документируем код эффективно при помощи Doxygen
<https://habrahabr.ru/post/252101/>
3. Настройка среды непрерывного развертывания с помощью Jenkins.
<https://www.ibm.com/developerworks/ru/library/d-continuous-delivery-framework-jenkins/index.html>
4. Jenkins – инструмент для непрерывной интеграции
<https://www.slideshare.net/alexgrakoff5/jenkins-54143962>
5. Arnold Robbins, Bash Reference Card, SSC, 1998, 1-58731-010-5
6. Arnold Robbins, Effective Awk Programming, Free Software Foundation / O'Reilly and Associates, 2000, 1-882114-26-4.

Организация управляемой самостоятельной работы

Основными направлениями управляемой самостоятельной работы в овладении знаниями учебной дисциплины «Автоматизация проектирования программного обеспечения» являются:

- изучение материалов лекций и слайдов, размещенных на образовательном ресурсе университета;
- ознакомление с программой учебной дисциплины;
- ознакомление со списком рекомендуемой литературы по дисциплине в целом, изучение необходимой литературы по теме, подбор дополнительной литературы в сети интернет;
- получение навыков разработки в ходе выполнения самостоятельных практических заданий, а также выполнения тестов контроля знаний, размещенных в сети интернет;
- подготовка к зачету.

Материалы и задания, размещенные в сети интернет, подразумевают в дальнейшем переход на полное самостоятельное изучение материала, согласно главам в плане. В этом случае студент сможет получить знания главным образом за счет самостоятельной работы.

Перечень используемых средств диагностики результатов учебной деятельности

С целью текущего контроля знаний студентов предусматривается проведение собеседования, контрольных, самостоятельных работ и выполнения тестовых заданий. По итогам обучения проводится зачет.

Рекомендуемый перечень заданий УСР

1. Создание скрипта на языке bash, с обработкой входных параметров
2. Обработка текстовой информации с использованием регулярных выражений, утилит sed, awk.
3. Создание сценария запуска компиляции файлов C++ с использованием make.
4. Создание сценария непрерывной интеграции в Jenkins.
5. Генерация html документации с помощью Doxygen.

Методика формирования итоговой оценки

Итоговая оценка формируется на основе трех документов:

1. Правила проведения аттестации (Постановление №53 от 29.05.2012г.)
2. Положение о рейтинговой системе БГУ (ред.2015г.)
3. Критерии оценки студентов (10 баллов).

Рекомендуемый перечень заданий лабораторных занятий

1. Резервное архивирование всех файлов, которые были изменены в течение последних суток.
2. `wh, whois`, выяснение имени домена.
3. Проверка "битых" ссылок.
4. Проверка - является ли строка пустой.
5. Вставка пустых строк между параграфами в текстовом файле.
6. Создание списка аргументов в цикле `for` с помощью операции подстановки команд.
7. Вывести список всех пользователей системы.
8. Проверка авторства всех бинарных файлов в текущем каталоге.
9. Вывести список символических ссылок в каталоге.
10. Вывести список символических ссылок в каталоге, сохраняемый в файле.
11. Создание меню с помощью `case`.
12. Создание меню с помощью `select` в функции.
13. Удалить файлы в текущем каталоге, имена которых содержат недопустимые символы и пробелы
14. Удаление комментариев из файла с текстом программы на языке C
15. Поиск файла программы по идентификатору процесса
16. Удаление временного файла при нажатии на `Control-C`

Рекомендуемый перечень вопросов к зачету по дисциплине «Автоматизация проектирования программного обеспечения»

1. Вложенные блоки и перенаправление ввода-вывода.
2. Сохранение результата исполнения вложенного блока в файл.
3. Запуск цикла в фоновом режиме.
4. Присваивание значений переменным и подстановка значений переменных.
5. Простое присваивание.
6. Присваивание значений переменным простое и замаскированное.
7. Позиционные параметры.
8. Использование команды `shift`.
9. Экранированные символы.
10. Завершение / код завершения.
11. Использование символа `!` для логической инверсии кода возврата.
12. Эквиваленты команды `test -- /usr/bin/test, [], и /usr/bin/[`.
13. Арифметические выражения внутри `(())`.
14. Операции сравнения.
15. Арифметические операции.
16. Построение сложных условий, использующих `&&` и `||`.
17. Различные представления числовых констант.

18. \$IFS и пробельные символы.
19. Ограничения времени ожидания ввода.
20. Ограничение времени ожидания команды read.
21. Вывод списка аргументов с помощью переменных \$* и \$@.
22. Содержимое \$* и \$@, когда переменная IFS -- пуста.
23. Альтернативный способ извлечения подстрок.
24. Подстановка параметров и сообщения об ошибках.
25. Подстановка параметров и сообщение о "порядке использования".
26. Длина переменной.
27. Поиск по шаблону в подстановке параметров.
28. Поиск по шаблону при анализе произвольных строк.
29. Поиск префиксов и суффиксов с заменой по шаблону.
30. Объявление переменных с помощью инструкции declare.
31. Косвенные ссылки.
32. Передача косвенных ссылок в awk.
33. Генерация случайных чисел.
34. Получение псевдослучайных чисел с помощью awk.
35. Работа с переменными в стиле языка C.
36. Простой цикл for.
37. Цикл for с двумя параметрами в каждом из элементов списка.
38. Обработка списка файлов в цикле for.
39. Цикл for без списка аргументов.
40. Простой цикл while.
41. Цикл until.
42. Команды break и continue в цикле.
43. Прерывание многоуровневых циклов.
44. Передача управление в начало внешнего цикла.
45. Использование case.
46. Простой пример сравнения строк.
47. Ввод значений переменных (с помощью read).
48. Ввод многострочного текста (с помощью read).
49. Обнаружение нажатия на курсорные клавиши.
50. Смена текущего каталога.
51. Команда let, арифметические операции.
52. Принудительное завершение сеанса.
53. Установка значений аргументов с помощью команды set.
54. Изменение значений позиционных параметров (аргументов).
55. "Сброс" переменной.
56. Передача переменных во вложенный сценарий awk, с помощью export.
57. Прием опций/аргументов, передаваемых сценарию, с помощью getopts.
58. "Подключение" внешнего файла.
59. Команда exec.
60. Ожидание завершения процесса перед тем как продолжить работу.

61. Сценарий, завершающий себя сам с помощью команды `kill`.
62. Команда `date`.
63. Мониторинг системного журнала с помощью `tail`.
64. Преобразование символов в верхний регистр.
65. Изменение имен всех файлов в текущем каталоге в нижний регистр.
66. Преобразование текстового файла из формата DOS в формат UNIX.
67. Нумерация строк сценария.
68. Утилиты `basename` и `dirname`.
69. Использование `seq` для генерации списка аргументов цикла `for`.
70. Использование `getopt` для разбора аргументов командной строки.
71. Захват нажатых клавиш.
72. Надежное удаление файла.
73. Перенаправление `stdin` с помощью `exec`.
74. Перенаправление `stdout` с помощью `exec`.
75. Вывод многострочных сообщений с помощью `cat`.
76. Вывод многострочных сообщений с подавлением символов табуляции.
77. Отключение подстановки параметров.
78. Блочный комментарий.
79. Встроенная справка к сценарию.
80. Область видимости переменных.
81. Запуск нескольких процессов в подболочках.
82. Функция с аргументами.
83. Вывод цветного текста.
84. Утилита для переименования файлов.
85. Вывод дерева каталогов.
86. Функции для работы со строками.

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ
К УЧЕБНОЙ ПРОГРАММЕ УВО ПО ИЗУЧАЕМОЙ УЧЕБНОЙ
ДИСЦИПЛИНЕ на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
«Математической кибернетики» (протокол № от 201__ г.)

Заведующий кафедрой
д-р ф.-м. н., профессор

(подпись)

А.Л. Гладков

УТВЕРЖДАЮ
Декан факультета
к. ф.-м. н., доцент

(подпись)

Д.Г. Медведев