Курсовой проект по курсам
«Архитектура компьютеров» и «Программные и аппаратные средства информатики»:
8 факультет, 1 курс, осенний семестр 2011/12 учебного года

Цель: *Самостоятельное изучение конкретных вычислительных машин, комплексов, систем и сетей с оформлением технической документации.*

Задание I. Составить схему домашнего компьютера студента со всеми внутренними и внешними (периферийными) устройствами в окружении локальных/глобальных сетей. Формат схемы — А3 (от руки) или А4 (принтер с разрешением 300 dpi). К схеме должна прилагаться иллюстрированная (рисунками, схемами, фотографиями, таблицами) пояснительная записка — подробное (≈10–15 стр.) архитектурное (с точки зрения программиста!) описание аппаратных и программных средств. В пояснительной записке необходимо проанализировать отличия аппаратных и программных средств настольной ПЭВМ от крупномасштабных серверов и рабочих станций. В качестве объекта описания также может использоваться полнофункциональный переносной компьютер: лаптоп, ноутбук, нетбук, смартфон или коммуникатор.

Студенты, не имеющие домашнего компьютера, описывают тот, на котором они реально работали: школьный, соседа, друга, на работе родителей, либо (в будущем времени!) компьютер, который хотели бы иметь. В последнем случае необходимо предоставить справки об отсутствии компьютера во всех вышеупомянутых местах :-). В любом случае полезнее описывать реальный компьютер, нежели фантазировать.

Задание II. Составить схему лабораторной вычислительной системы с пояснительной запиской о её составе и функционировании. Использовать материалы лабораторных работ 1–4, в том числе схему сети и таблицу характеристик ЭВМ, данные операционной системы (dmesg, pciconfig) и результаты рекогносцировки на местности. Необходимые сведения можно найти на дисках CD-хрестоматии, в Интернете (см., например, www.ixbt.com, www.computer.org и др.). В пояснительную записку обязательно следует включить сравнительную характеристику используемых в лаборатории версий ОС UNIX.

Введение к КП (1–2 стр.) должно содержать цели и задачи КП (на языке технических требований (заданий) в будущем времени и с оттенком долженствования: *следует*, *надо*, …).

Теоретическая часть КП представляет собой *беллетристическое задание (эссе)* в форме обзора, реферата, теоретического исследования, толкового словаря терминов, биографии учёного или статьи энциклопедического характера по одному из устройств ЭВМ или о компонентах системного программного обеспечения. Задание должно быть связано с изучаемым курсом, в частности, использовать соответствующую терминологию. Примерные темы заданий приведены в текущем издании пособия по практикуму I семестра. Выбранная тема должна быть согласована с преподавателем и зафиксирована им в учётной карточке текущей успеваемости группы (кондуите).

Составляя отчёт по заданию, необходимо корректно цитировать и декларировать источники заимствований (кавычки, прямая ссылка на список использованных источников в квадратных скобках, или обороты-преамбулы типа «Как пишет Н. Вирт в своей книге „Систематическое программирование“, …» или «… следуя К. Шеннону, будем считать …»). За плагиатом и компиляцией не следует диссертация: преподаватель может заменить тему (задачу), дать дополнительное задание или назначить зачёт по сомнительному материалу, и, конечно же, снизить оценку.

Рекомендуемый обьём отчётов по каждому из заданий КП — 10–15 страниц.

В заключении (2-3 стр.) подводятся итоги выполнения заданий КП, сдачи зачетов и самостоятельной работы, делаются выводы: *сделано*, *получено*, *разработано*, *изучено*, *освоено* и т.п.

Схемы сети или компьютера можно сделать в какой-либо системе иллюстративной графики (MS Visio, Corel Draw!, GnomeDia и т. д.) или в чертёжно-графической системе (AutoCAD, Kredo), набрать вручную в формате PostScript или выполнить от руки, если нет навыков работы в таких системах. Однако, не стоит тратить много времени на изготовление такой схемы!

Оценка за каждое задание выставляется по итогам проверки и защиты с учётом сроков фактического выполнения. Сроки сдачи устанавливаются после изложения на лекциях соответствующего материала. Оценка за проект в целом определяется как средняя по всем пяти заданиям также с учётом времени сдачи . Оценка выставляется в зачётку и в ведомость одновременно только после представления полного отчёта по КП в скоросшивателе или депонирования электронного документа. Споры по оцениванию КП разрешаются комиссией с участием лектора курса в соответствии с действующей инструкцией об оценке знаний.

Практикум по курсам «Основы информатики», «Алгоритмы и структуры данных».

Курсовой проект: 8 факультет, I курс, I семестр 2011/12 уч. года.

Задание III. Вещественный тип. Приближенные вычисления. Табулирование функций

Составить программу на Си, которая печатает таблицу значений элементарной функции, вычисленной двумя способами: по формуле Тейлора и с помощью встроенных функций языка программирования. В качестве аргументов таблицы взять точки разбиения отрезка [a,b] на n равных частей (n + 1 точка включая концы отрезка), находящихся в рекомендованной области хорошей точности формулы Тейлора. Вычисления по формуле Тейлора проводить по экономной в сложностном смысле схеме с точностью $\varepsilon * k$, где ε – машинное эпсилон аппаратно реализованного вещественного типа для данной ЭВМ, а k – экспериментально подбираемый коэффициент, обеспечивающий приемлемую сходимость. Число итераций должно ограничиваться сверху числом порядка 100. Программа должна сама определять машинное ε и обеспечивать корректные размеры генерируемой таблицы.

Дополнительное задание

Для углубленного изучения вещественных типов рекомендуется провести вычисление машинного эпсилон для других (нестандартных) разновидностей вещественных типов на DEC Alpha, а также, по возможности, для других систем программирования и аппаратных средств. Сравните полученные результаты со встроенными константами системы программирования.

Для изучения атрибутов вещественного и целого типов определите границы допустимого диапазона значений программным путем и сравните с соответствующими константами. Объясните полученные результаты.

Дополнительное задание оформляется в виде отдельных программ.

Полученные результаты необходимо включить в отчет по курсовому проекту. Успешное выполнение дополнительного задания учитывается при оценке основного задания.

Замечание. Формула Тейлора сводит вычисление трансцендентных функций к алгебраическим (полиномам; схему Горнера – в студию!). Однако этот простой способ не применяется на практике ввиду большой ресурсоёмкости и значительной погрешности. Изучение более совершенных способов вычисления значений трансцендентных функций на ЭВМ производится в курсе численных методов.

Пример результатов для sin(x) = $\sum_{n=0}^{N}(-1)^n \frac{x^{2n+1}}{(2n+1)!}$

```
Машинное эпсилон для типа long double в системе Compaq C на Digital Alpha = ...
            Таблица значений ряда Тейлора и стандартной функции для f(x)=sin x
```

x	**част. сумма ряда для sin x**	**значения функции sin x**	**число итераций**
0.00	**...**	**0.0**	**...**
0.05		**0.0008 …**	
0.10		**0.0017 …**	
0.15		**0.0026 …**	
...	**...**	**...**	**...**
0.50	**...**	**...**	**...**

Варианты заданий

№	**ряд**	**a**	**b**	**функция**
1	$\frac{x}{9}-\frac{x^3}{9^2}+\ldots+(-1)^n\frac{x^{2n+1}}{9^{n+1}}$	–1.0	1.0	$\frac{x}{9+x^2}$
2	$2(\frac{x}{1}+\frac{x^3}{3}+\ldots+\frac{x^{2n+1}}{2n+1})$	0.0	0.5	$\ln\frac{1+x}{1-x}$
3	$x-\frac{5}{2}x^2+\ldots+\frac{(-1)^{n+1}\cdot 2^n-1}{n}x^n$	–0.2	0.3	$\ln(1+x-2x^2)$
4	$\ln 2+\frac{x}{2}-\frac{x^2}{2^3}+\ldots+(-1)^{n-1}\frac{x^n}{n\cdot 2^n}$	–1.0	1.0	$\ln(2+x)$
5	$-\frac{4x^2}{2}+\frac{16x^4}{24}+\ldots+(-1)^n\frac{(2x)^{2n}}{(2n)!}$	0.0	0.5	$2(\cos^2 x-1)$
6	$x+\frac{x^3}{3!}+\ldots+\frac{x^{2n-1}}{(2n-1)!}$	0.0	1.0	$\operatorname{sh} x$

7	$3x+8x^2+\ldots+n\cdot(n+2)x^n$	0.0	0.5	$\dfrac{x(3-x)}{(1-x)^3}$
8	$-\dfrac{1}{5}-\dfrac{2x}{5^2}-\dfrac{4x^2}{5^3}-\ldots-\dfrac{2^{n-1}x^{n-1}}{5^n}$	0.0	2.0	$\dfrac{1}{2x-5}$
9	$-(1+\dfrac{2}{3})-(1+\dfrac{2}{3^2})x-\ldots-(1+\dfrac{2}{3^{n+1}})x^n$	0.0	0.5	$\dfrac{3x-5}{x^2-4x+3}$
10	$\dfrac{2x^2}{2!}-\dfrac{2^3x^4}{4!}+\ldots+(-1)^{n-1}\dfrac{2^{2n-1}x^{2n}}{(2n)!}$	0.0	1.0	$\sin^2 x$
11	$1-\dfrac{3}{2}x^2+\ldots+(-1)^n\dfrac{2n^2+1}{(2n)!}x^{2n}$	0.1	0.6	$(1-\dfrac{x^2}{2})\cos x-\dfrac{x}{2}\sin x$
12	$1+\dfrac{\ln 3}{1!}x+\dfrac{\ln^2 3}{2!}x^2+\ldots+\dfrac{\ln^n 3}{n!}x^n$	0.0	1.0	3^x
13	$x-\dfrac{x^3}{3!}+\ldots+(-1)^n\dfrac{x^{2n+1}}{(2n+1)!}$	0.0	1.0	$\sin x$
14	$-3-4x-5x^2-\ldots-(n+3)x^n$	0.1	0.6	$\dfrac{2x-3}{(x-1)^2}$
15	$1-\dfrac{x^2}{2!}+\ldots+(-1)^n\dfrac{x^{2n}}{(2n)!}$	0.0	1.0	$\cos x$
16	$1+3x^2+\ldots+\dfrac{2n+1}{n!}x^{2n}$	0.0	1.0	$(1+2x^2)e^{x^2}$
17	$\dfrac{x-1}{x+1}+\dfrac{1}{3}(\dfrac{x-1}{x+1})^3+\ldots+\dfrac{1}{2n+1}(\dfrac{x-1}{x+1})^{2n+1}$	0.2	0.7	$\dfrac{1}{2}\ln x$
18	$\dfrac{x^3}{3}-\dfrac{x^5}{15}+\ldots+(-1)^{n+1}\dfrac{x^{2n+1}}{4n^2-1}$	0.1	0.6	$\dfrac{1+x^2}{2}\operatorname{arctg} x-\dfrac{x}{2}$
19	$1+\dfrac{x^2}{2}+\ldots+\dfrac{x^{2n}}{(2n)!}$	0.1	0.6	$\operatorname{ch} x$
20	$1+\dfrac{2x}{1!}+\ldots+\dfrac{(2x)^n}{n!}$	0.1	0.6	e^{2x}
21	$1+2\dfrac{x}{2}+\ldots+\dfrac{n^2+1}{n!}(\dfrac{x}{2})^n$	0.1	0.6	$(\dfrac{x^2}{4}+\dfrac{x}{2}+1)e^{\frac{x}{2}}$
22	$1-\dfrac{x^2}{2}+\dfrac{x^3}{3}+\ldots+(-1)^{n-1}\dfrac{n-1}{n!}x^n$	0.0	1.0	$(1+x)e^{-x}$
23	$x-\dfrac{x^3}{3}+\ldots+(-1)^n\dfrac{x^{2n+1}}{2n+1}$	0.0	0.5	$\operatorname{arctg} x$
24	$1+\dfrac{x^2}{1}+\dfrac{x^4}{2}+\ldots+\dfrac{x^{2n}}{n!}$	0.0	1.0	e^{x^2}
25	$\dfrac{1}{4}+\dfrac{x^4}{4^2}+\ldots+\dfrac{x^{4n}}{4^{n+1}}$	0.0	1.0	$\dfrac{1}{4-x^4}$
26*	$-\cos x+\dfrac{\cos 2x}{2^2}+\ldots+(-1)^n\dfrac{\cos nx}{n^2}$	$\dfrac{\pi}{5}$	π	$\dfrac{1}{4}(x^2-\dfrac{\pi^2}{3})$
27*	$1+\dfrac{\cos x}{1!}+\ldots+\dfrac{\cos nx}{n!}$	0.1	0.6	$e^{\cos x}\cdot\cos(\sin x)$
28*	$\cos x+\dfrac{\cos 2x}{2}+\ldots+\dfrac{\cos nx}{n}$	$\dfrac{\pi}{5}$	$\dfrac{6\pi}{5}$	$-\ln\lvert 2\sin\frac{x}{2}\rvert$

Задание составил к.ф.-м.н., доц. Зайцев В.Е. Варианты подготовлены к.ф.-м.н., доц. Сопруненко И.П.

Практикум по курсам «Основы информатики», «Алгоритмы и структуры данных».

Курсовой проект: 8 факультет, I курс, I семестр 2011/12 уч. года.

Задание IV: Процедуры и функции в качестве параметров

Составить программу на языке Си с процедурами решения трансцендентных алгебраических уравнений различными численными методами (итераций, Ньютона и половинного деления — дихотомии). Нелинейные уравнения оформить как параметры-функции, разрешив относительно неизвестной величины в случае необходимости. Применить каждую процедуру к решению двух уравнений, заданных двумя строками таблицы, начиная с варианта с заданным номером. Если метод неприменим, дать математическое обоснование и графическую иллюстрацию, например, с использованием `gnuplot`. Классические примеры использования процедур и функций в качестве параметров см. в [8], п. 12.4, и в [10], п. 9.2.

Краткие сведения из численных методов

Рассматривается уравнение вида $F(x)=0$. Предполагается, что функция $F(x)$ достаточно гладкая, монотонная на этом отрезке и существует единственный корень уравнения $x^* \in [a,b]$. На отрезке $[a,b]$ ищется приближенное решение *x* с точностью *ε*, т.е. такое, что $|x-x^*|<\varepsilon$.

При решении реальных задач, где поведение функции $F(x)$ неизвестно, сначала производят исследование функции (аналитическое, численное, или графическое (`gnuplot, MathLab, MathCAD, Maple`)) и т. н. отделение корней, т. е. разбивают область определения функции на отрезки монотонности, на каждом из которых имеется ровно один корень и выполняются другие условия применимости численных методов (гладкость). Различные численные методы предъявляют разные требования к функции $F(x)$, обладают различной скоростью сходимости и поведением.

В данном задании предлагается изучить и запрограммировать три простейших численных метода решения алгебраических уравнений и провести вычислительные эксперименты по определению корней уравнений на указанных в задании отрезках монотонности и, в качестве дополнительного упражнения, вне их.

1. Метод дихотомии (половинного деления).

Очевидно, что если на отрезке $[a,b]$ существует корень уравнения, то значения функции на концах отрезка имеют разные знаки: $F(a)\cdot F(b)<0$. Метод заключается в делении отрезка пополам и его сужении в два раза на каждом шаге итерационного процесса в зависимости от знака функции в середине отрезка.

Итерационный процесс строится следующим образом: за начальное приближение принимаются границы исходного отрезка $a^{(0)}=a$, $b^{(0)}=b$. Далее вычисления проводятся по формулам: $a^{(k+1)}=(a^{(k)}+b^{(k)})/2$, $b^{(k+1)}=b^{(k)}$, если $F(a^{(k)})\cdot F((a^{(k)}+b^{(k)})/2)>0$; или по формулам: $a^{(k+1)}=a^{(k)}$, $b^{(k+1)}=(a^{(k)}+b^{(k)})/2$, если $F(b^{(k)})\cdot F((a^{(k)}+b^{(k)})/2)>0$.

Процесс повторяется до тех пор, пока не будет выполнено условие окончания $|a^{(k)}-b^{(k)}|<\varepsilon$.

Приближенное значение корня к моменту окончания итерационного процесса получается следующим образом $x^*\approx(a^{(конечное)}+b^{(конечное)})/2$.

2. Метод итераций.

Идея метода заключается в замене исходного уравнения $F(x)=0$ уравнением вида $x=f(x)$.

Достаточное условие сходимости метода: $|f'(x)|<1, x\in[a,b]$. Это условие необходимо проверить перед началом решения задачи, так как функция $f(x)$ может быть выбрана неоднозначно, причем в случае неверного выбора указанной функции метод расходится.

Начальное приближение корня: $x^{(0)}=(a+b)/2$ (середина исходного отрезка).

Итерационный процесс: $x^{(k+1)}=f(x^{(k)})$.

Условие окончания: $|x^{(k)}-x^{(k-1)}|<\varepsilon$.

Приближенное значение корня: $x^*\approx x^{(конечное)}$.

3. Метод Ньютона.

Метод Ньютона является частным случаем метода итераций.

Условие сходимости метода: $|F(x)\cdot F''(x)|<(F'(x))^2$ на отрезке $[a,b]$.

Итерационный процесс: $x^{(k+1)}=x^{(k)}-F(x^{(k)})/F'(x^{(k)})$.

Более совершенное с программистской точки зрения решение задачи может быть получено с помощью изучаемого в курсе «Языки программирования» (II семестр) процедурного типа данных. В этом случае различные уравнения и методы как переменные процедурного типа подставляются в качестве фактических параметров соответствующих подпрограмм. Решение задачи на языке Си, фактически базирующееся на указателях на функции, близко к этому.

Варианты заданий (составлены к.ф.-м.н., доц. Сопруненко И.П.):

№	Уравнение	Отрезок, содержащий корень	Базовый метод	Приближенное значение корня
1	$e^x + \ln x - 10x = 0$	[3, 4]	Ньютона	3.5265
2	$\cos x - e^{-\frac{x^2}{2}} + x - 1 = 0$	[1, 2]	дихотомии	1.0804
3	$1 - x + \sin x - \ln(1 + x) = 0$	[1, 1.5]	итераций	1.1474
4	$3x - 14 + e^x - e^{-x} = 0$	[1, 3]	Ньютона	2.0692
5	$\sqrt{1 - x} - \mathrm{tg} x = 0$	[0, 1]	дихотомии	0.5768
6	$x + \cos(x^{0,52} + 2) = 0$	[0.5, 1]	итераций	0.9892
7	$3\ln^2 x + 6\ln x - 5 = 0$	[1, 3]	Ньютона	1.8832
8	$0{,}6 \cdot 3^x - 2{,}3x - 3 = 0$	[2, 3]	дихотомии	2.4200
9	$x^2 - \ln(1 + x) - 3 = 0$	[2, 3]	итераций	2.0267
10	$2x \cdot \sin x - \cos x = 0$	[0.4, 1]	Ньютона	0.6533
11	$e^x + \sqrt{1 + e^{2x}} - 2 = 0$	[-1, 0]	дихотомии	-0.2877
12	$\ln x - x + 1{,}8 = 0$	[2, 3]	итераций	2.8459
13	$x \cdot \mathrm{tg}\, x - \frac{1}{3} = 0$	[0.2, 1]	Ньютона	0.5472
14	$\mathrm{tg}\frac{x}{2} - \mathrm{ctg}\frac{x}{2} + x = 0$	[1, 2]	дихотомии	1.0769
15	$0{,}4 + \mathrm{arctg}\sqrt{x} - x = 0$	[1, 2]	итераций	1.2388
16	$3\sin\sqrt{x} + 0.35x - 3.8 = 0$	[2, 3]	итераций	2.2985
17	$0{,}25x^3 + x - 1{,}2502 = 0$	[0, 2]	Ньютона	1.0001
18	$x + \sqrt{x} + \sqrt[3]{x} - 2{,}5 = 0$	[0.4, 1]	дихотомии	0.7376
19	$x - \frac{1}{3 + \sin 3.6x} = 0$	[0, 0.85]	итераций	0.2624
20	$0{,}1x^2 - x\ln x = 0$	[1, 2]	Ньютона	1.1183
21	$\mathrm{tg}\, x - \frac{1}{3}\mathrm{tg}^3 x + \frac{1}{5}\mathrm{tg}^5 x - \frac{1}{3} = 0$	[0, 0.8]	дихотомии	0.3333
22	$\arccos x - \sqrt{1 - 0{,}3x^3} = 0$	[0, 1]	итераций	0.5629
23	$3x - 4\ln x - 5 = 0$	[2, 4]	Ньютона	3.23
24	$\cos\frac{2}{x} - 2\sin\frac{1}{x} + \frac{1}{x} = 0$	[1, 2]	дихотомии	1.8756
25	$\sqrt{1 - 0{,}4x^2} - \arcsin x = 0$	[0, 1]	итераций	0.7672
26	$e^x - e^{-x} - 2 = 0$	[0, 1]	Ньютона	0.8814
27	$\sin(\ln x) - \cos(\ln x) + 2\ln x = 0$	[1, 3]	дихотомии	1.3749
28	$x - 2 + \sin\frac{1}{x} = 0$	[1.2, 2]	итераций	1.3077

Замечание: для вычислений и проверки сходимости методов в разных уравнениях каждого задания должно применяться как аналитическое, так и численное (конечноразностное) дифференцирование функций. Кроме того, может быть полезным предусмотреть в программе проверку полученного корня его подстановкой в уравнение F(x) = 0. Образующаяся при этом невязка может служить ещё одним критерием качества приближённого решения.

Полезно также посетить интерактивный решатель задач по численным методам, разработанный аспирантом каф. 806 Кичинским К.А. под руководством проф. Ревизникова Д.Л.: http://www.nummeth.mainfo.ru/