



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ

С.В. Дианов

АРХИТЕКТУРА КОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Учебно-методическое пособие
по проведению практических занятий

для студентов II курса
направления 09.03.01
очной формы обучения

Москва · 2022

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ВОЗДУШНОГО ТРАНСПОРТА
(РОСАВИАЦИЯ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ГРАЖДАНСКОЙ АВИАЦИИ» (МГТУ ГА)

Кафедра вычислительных машин, комплексов, систем и сетей

С.В. Дианов

АРХИТЕКТУРА КОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Учебно-методическое пособие
по проведению практических занятий

*для студентов II курса
направления 09.03.01
очной формы обучения*

Москва
ИД Академии Жуковского
2022

УДК 681.324
ББК 6Ф7.3
Д44

Рецензент:

Овсянникова Н.И. – канд. физ.-мат. наук

Дианов С.В.
Д44 **Дианов С.В.** Архитектура компьютерных вычислительных систем [Текст] : учебно-методическое пособие по проведению практических занятий / С.В. Дианов. – М.: ИД Академии Жуковского, 2022. – 28 с.

Данное учебно-методическое пособие издается в соответствии с рабочей программой учебной дисциплины «Архитектура компьютерных вычислительных систем» по учебному плану направления подготовки 09.03.01 для студентов II курса очной формы обучения.

Рассмотрено и одобрено на заседаниях кафедры 25.01.2022 г. и методического совета 25.01.2022 г.

УДК 681.324
ББК 6Ф7.3

В авторской редакции

Подписано в печать 29.06.2022 г.
Формат 60x84/16 Печ. л. 1,75 Усл. печ. л. 1,63
Заказ № 889/0603-УМП06 Тираж 30 экз.

Московский государственный технический университет ГА
125993, Москва, Кронштадтский бульвар, д. 20

Издательский дом Академии имени Н. Е. Жуковского
125167, Москва, 8-го Марта 4-я ул., д. 6А
Тел.: (495) 973-45-68
E-mail: zakaz@itsbook.ru

© Московский государственный технический университет гражданской авиации, 2022

Практическое занятие №1

Кодирование печатных символов в цифровом компьютере

Цель и задачи проведения практического занятия.

Ознакомиться с основными системами кодирования информации в цифровом компьютере. Овладеть методикой использования кодировочных таблиц. Получить практические навыки использования наиболее распространённых систем кодирования.

Методические указания по теме. Краткие теоретические сведения.

Рассмотрим наиболее широко распространённые системы кодирования.

ASCII —American standard code for information interchange. Это таблица, в которой содержатся числовые коды, сопоставленные печатным и непечатным символам.

Данная таблица была разработана в США, там же она и была стандартизирована, в 1963 году. Первоначально таблица была разработана для кодирования символов с кодами, которые вмещались в 7 бит, то есть имели 128 символов или 2^7 . Старшим битом являлся седьмой - он использовался для проверки и контроля ошибок, которые могли возникнуть при передаче данных. Нумерация битов начиналась с 0, поэтому седьмой бит был незначимым, а по счёту восьмым. Весь код занимал 1 байт - 8 разрядов. Изначально таблицей кодировались только заглавные буквы, но также предполагался резерв, представляющий из себя группы по 16 символов, которые резервировались для их дальнейшего расширения в область для строчных букв или же область для управляющих символов. В дальнейшем, после изменений 1967 года, кодировка таблицы была расширена до 256 символов, коды изначальных 128 символов не изменились. Эта улучшенная версия ASCII стала восприниматься как половина 8 битной кодировки.

Национальные варианты.

Стандарт ISO 64 Предусматривает возможность размещения в ASCII национальных символов, путём замены определенных символов на другие. Эта система подходит для языков Европы. Для некоторых языков, на пример русского, существуют более радикальные модификации, которые разрешали заменять строчные латинские буквы на национальные символы. Вариант ASCII, без каких-либо изменений и национальных символов, называется US-ASCII. Удобнее всего оказалось использовать 8-битовые кодировки, в которых нижняя

часть кодовой таблицы от 0 до 127 символа занимают символы US-ASCII, а верхнюю – дополнительные с национальными символами.

С помощью ASCII можно закодировать:

- знаки препинания;
- управляющие символы;
- десятичные цифры;
- буквы национального алфавита;
- буквы латинского алфавита.

Таблица 1. Код ASCII (фрагмент)

32 пробел	48 0	64 @	80 P	96 `	112 p
33	49 1	65 A	81 Q	97 a	113 q
34 "	50 2	66 B	82 R	98 b	114 r
35 #	51 3	67 C	83 S	99 c	115 s
36 \$	52 4	68 D	84 T	100 d	116 t
37 %	53 5	69 E	85 U	101 e	117 u
38 &	54 6	70 F	86 V	102 f	118 v
39 ' .	55 7	71 G	87 W	103 g	119 w
40 (56 8	72 H	88 X	104 h	120 x
41)	57 9	73 I	89 Y	105 i	121 y
42 *	58 :	74 J	90 Z	106 j	122 z
43 +	59 ;	75 K	91 [107 k	123 {
44 ,	60 <	76 L	92 \	108 l	124
45 -	61 =	77 M	93]	109 m	125 }
46 .	62 >	78 N	94 ^	110 n	126 ~
47 /	63 ?	79 O	95 _	111 o	127

Пример :

72 69 108 108 111 32 119 111 114 108 100 33 – Hello world!

Расширенные символы ASCII Win-1251 кириллица (код символа 128-255) можно найти в интернете, например здесь <https://www.industrialnets.ru/files/misc/ascii.pdf>

Unicode— это кодировка, включающая в себя знаки практически всех письменных языков мира. На данный момент является самым распространенным стандартом.

История происхождения. Данный стандарт был предложен в 1991 году, идея принадлежала организации Консорциум Юникод. Применения этого стандарта подразумевало возможность кодировать очень большое число символов из различных систем письменности, то есть система объединяет, как мировые языки, так и математические символы, символы музыкальной нотной нотации. Таким образом появляется преимущество в ненужности переключения кодовых страниц. Стандарт имеет две основные части: это универсальный набор символов и семейство кодировок. Первая часть отвечает за перечисление допустимых по стандарту Юникод символов, а также присваивает каждому символу код в виде неотрицательного целого числа, оно записывается в шестнадцатеричной форме, имеет префикс U+. Вторая его часть определяет способы преобразования символов для передачи в потоке или файле. Так как Unicode является до сих пор самым распространенным стандартом, то его развитие продолжается и по сей день, добавление различных дополнительных символов. К примеру, в 2020 году в систему были добавлены: хорезмийское письмо, письмо дивес акуру, езидское письмо, различные эмодзи и т.д.

Кодовое пространство. Кодовое пространство состоит из 17 плоскостей по 65536 символов. Нулевая плоскость – базовая, содержит символы наиболее употребительных письменностей. Все остальные плоскости являются дополнительными.

Система кодирования. Юникод представляет собой набор графических символов и способы их кодирования для компьютерной обработки текстовых данных.

Общие принципы кодирования: гарантии стабильности, динамическая компоновка, логический порядок, преобразуемость, семантика, универсальность, унификация, эффективность, политика консорциума.

Способы представления. Юникод имеет различные формы его представления: UTF-8, UTF-16 и UTF-32. Также была разработана форма UTF-7, но из-за несовместимости с ASCII – не была распространена и не включена в стандарт.

Коды разделены на несколько областей. Область от U+0000 до U+007F содержит символы набора ASCII, а также коды этих символов совпадают с их кодами в ASCII. Часть кодов зарезервирована для использования в будущем. Для символов кириллицы выделены области знаков с кодами от U+0400 до U+052F, от U+2DE0 до U+2DFF, от U+A640 до U+A69F.

Таблица 2 Код Unicode (фрагмент)

0020	0	0030	@	0040	P	0050	`	0060	p	0070	00A0	°	00B0	À	00C0	Ð	00D0	à	00E0	ð	00F0		
!	0021	1	0031	A	0041	Q	0051	a	0061	q	0071	ı	00A1	±	00B1	Á	00C1	Ñ	00D1	á	00E1	ñ	00F1
"	0022	2	0032	B	0042	R	0052	b	0062	r	0072	¢	00A2	²	00B2	Â	00C2	Ò	00D2	â	00E2	ò	00F2
#	0023	3	0033	C	0043	S	0053	c	0063	s	0073	£	00A3	³	00B3	Ã	00C3	Ó	00D3	ã	00E3	ó	00F3
\$	0024	4	0034	D	0044	T	0054	d	0064	t	0074	¤	00A4	´	00B4	Ä	00C4	Ô	00D4	ä	00E4	ô	00F4
%	0025	5	0035	E	0045	U	0055	e	0065	u	0075	¥	00A5	µ	00B5	Å	00C5	Õ	00D5	å	00E5	õ	00F5
&	0026	6	0036	F	0046	V	0056	f	0066	v	0076	¦	00A6	¶	00B6	Æ	00C6	Ö	00D6	æ	00E6	ö	00F6
'	0027	7	0037	G	0047	W	0057	g	0067	w	0077	§	00A7	·	00B7	Ç	00C7	×	00D7	ç	00E7	÷	00F7
(0028	8	0038	H	0048	X	0058	h	0068	x	0078	¨	00A8	,	00B8	È	00C8	Ø	00D8	è	00E8	ø	00F8
)	0029	9	0039	I	0049	Y	0059	i	0069	y	0079	©	00A9	¹	00B9	É	00C9	Ù	00D9	é	00E9	ù	00F9
*	002A	:	003A	J	004A	Z	005A	j	006A	z	007A	ª	00AA	º	00BA	Ê	00CA	Ú	00DA	ê	00EA	ú	00FA
+	002B	;	003B	K	004B	[005B	k	006B	{	007B	«	00AB	»	00BB	Ë	00CB	Û	00DB	ë	00EB	û	00FB
,	002C	<	003C	L	004C	\	005C	l	006C		007C	¬	00AC	¼	00BC	Ì	00CC	Ü	00DC	ì	00EC	ü	00FC
-	002D	=	003D	M	004D]	005D	m	006D	}	007D	-	00AD	½	00BD	Í	00CD	Ý	00DD	í	00ED	ý	00FD
.	002E	>	003E	N	004E	^	005E	n	006E	~	007E	®	00AE	¾	00BE	Î	00CE	Þ	00DE	î	00EE	þ	00FE
/	002F	?	003F	O	004F	_	005F	o	006F	˘	007F	¯	00AF	¿	00BF	Ï	00CF	ß	00DF	ï	00EF	ÿ	00FF

Пример:

```
\u+0048\u+0065\u+006c\u+006c\u+006f\u+0020\u+0077\u+006f\u+0072\u+
006c\u+0064\u+0021\u- Hello world!
```

КОИ-8 — это восьмибитовая кодовая страница, совместимая с ASCII. Была разработана для кодирования букв кириллических алфавитов. Используется для кодирования букв кириллических алфавитов. Таким образом была широко распространена как основная русская кодировка в системах Unix. На данный момент заменена Юникодом, как более популярным вариантом кодировки в сети.

История происхождения. Кодировка появилась в 1990 году, Ее автором является Андрей Чернов. На тот момент КОИ8 стала практически стандартом для кодирования русской кириллицы в Unix-подобных операционных системах, а также электронной почте.

В данной кодировке символы русского алфавита были помещены в верхней части кодовой таблицы таким образом, что позиции символов кириллицы соответствовали их фонетическим аналогам в английском алфавите и нижней части таблицы.

Принцип построения. Для каждого символа убирается по одному биту слева, следовательно получается относительно читаемый текст, который схож с транслитом - написание русских слов латинскими буквами. Символы кириллицы не располагаются в алфавитном порядке в таблице.

Способы представления. Существуют несколько вариантов кодировки КОИ8 в зависимости от кириллических алфавитов, общий диапазон 192-255 с 32 русскими буквами (не изменяется). Виды: КОИ8-U русско-украинский, КОИ8-RU включает и белорусский, КОИ8-C для средней Азии, и т.д. Латиница полностью соответствует кодировке ASCII.

Таблица 3 Код КОИ8 (фрагмент)

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.A	.B	.C	.D	.E	.F
8.	— 2500	 2502	┌ 250C	┐ 2510	└ 2514	┘ 2518	┑ 251C	┒ 2524	┓ 252C	└ 2534	┘ 253C	■ 2500	■ 2504	■ 2508	■ 250C	■ 2500
9.	▒ 2591	▒ 2592	▒ 2593	┌ 2520	■ 2540	· 2219	√ 2214	≈ 2248	≤ 2264	≥ 2265	∅ A0	∅ 2521	∅ B0	∅ B2	∅ B7	∅ F7
A.	= 2550	 2551	ƒ 2552	ё 451	г 2553	ф 2554	г 2555	г 2556	г 2557	л 2558	л 2559	л 255A	л 255B	л 255C	л 255D	л 255E
B.	 255F	 2560	 2561	Е 401	 2562	 2563	т 2564	т 2565	т 2566	л 2567	л 2568	л 2569	л 256A	л 256B	л 256C	© A9
C.	Ю 44E	а 430	б 431	ц 445	д 434	е 435	ф 444	г 433	х 445	и 438	й 439	к 43A	л 43B	м 43C	н 43D	о 43E
D.	п 43F	я 44F	р 440	с 441	т 442	у 443	ж 436	в 432	ь 44C	ы 448	з 437	ш 448	э 44D	щ 449	ч 447	ъ 44A
E.	Ю 42E	А 410	Б 411	Ц 425	Д 414	Е 415	Ф 424	Г 413	Х 425	И 418	Й 419	К 41A	Л 41B	М 41C	Н 41D	О 41E
F.	П 41F	Я 42F	Р 420	С 421	Т 422	У 423	Ж 416	В 412	Ь 42C	Ы 428	З 417	Ш 428	Э 42D	Щ 429	Ч 427	Ъ 42A

Пример:

412 441 435 43с 32 41F 440 438 423 435 442 Всем Привет (32- пробел, во фрагменте таблицы не указан)

Задание на практическое занятие:

1. Представить свою фамилию в рассмотренных системах кодирования.

Контрольные вопросы:

1. Какими основными кодами представляется информация в цифровом компьютере ?
2. Что такое ASCII ?
3. Каковы достоинства ASCII ?
4. Что такое UNICODE ?
5. Чем отличаются UNICODE и ASCII ?
6. Для чего разработан UNICODE ?
7. Что такое КОИ8 ?
8. Кто и для чего разработал КОИ8 ?

Практическое занятие №2

Стандарт IEEE754. Представление чисел с плавающей точкой.

Цель и задачи проведения практического занятия.

Знакомство с арифметикой выполнения вычислений с плавающей точкой. Изучение методики перевода вещественного числа в стандарт IEEE754. Получение практических навыков перевода числа в стандарт IEEE754.

Методические указания по теме. Краткие теоретические сведения.

Стандарт IEEE 754 (аббревиатура от Институт инженеров электротехники и электроники, разрабатывающий стандарты для радиоэлектроники) используется в компиляторах и процессорах для описания представления чисел с плавающей точкой. Стандарт описывает операции сложения, умножения, вычитания, деления, вычисления остатка от деления, извлечения квадратного корня, изменения типов представления чисел, исключительные случаи как деление на ноль.

Стандарт IEEE 754 определяет несколько возможных типов чисел с плавающей точкой (отличающихся числом бит, отводимых под порядок и

мантиссу): половинная точность (16 бит), одинарная точность (32 бита), четверная точность (128 бит) и расширенная точность (80 бит). Представления числа с плавающей точкой по стандарту содержит 3 части:

- Знак. Под знак всегда выделяется 1 бит, 0 используется для представления положительного числа, 1 напротив для отрицательного
- Смещённая экспонента. Число бит, выделяемое на смещение и мантиссу, определяется типом заданного числа
- Мантисса. Последняя часть представления числа, она представляет старшие разряды действительного числа

Рис.1 Схема представления числа 32 и 64 бита



У стандарта есть диапазон представления чисел, верхняя граница примерно равна $3.404 * 10^{38}$, нижняя граница $1.175 * 10^{-38}$

Разберем стандарт IEEE 754 на примере представления числа 0,03 с положительным и отрицательным знаком на 32 бита.

1) В первую очередь нам нужно определить степень для 2, произведем умножение числа на 2 столько раз, сколько нужно для преобразования в число с целой частью

$$0,03 * 2 = 0,06$$

$$0,06 * 2 = 0,12$$

...

$$0,96 * 2 = 1,92$$

$$1,92 * 2^{-6}, \text{ остаток } 0,92$$

2) Следующий этап умножение остатка на 2, произведем его 23 раза для записи мантиссы числа. В мантиссу записывается целая часть (0 или 1) числа после его умножения, дробная часть снова умножается на 2.

$$1) 0,92 * 2 = 1,84 \rightarrow 1$$

...

$$9) 0,04 * 2 = 0,08 \rightarrow 0$$

$$\dots$$

$$23) 0,36 * 2 = 0,72$$

3) Теперь определим смещение, ссылаясь на степень 2 первого этапа. Учитываем, что 1 бит определен для числа знака.

Порядок смещения: $127 - 6 = 121$

Переведем 121 из десятичной в двоичную, смещение $121 = 1111001$

0	01111001	11101011100001010001111
^	^	^
Знак	Экспонента	Мантисса

Мы записали 0,03 по стандарту, теперь запишем его же с отрицательным знаком. Запись отличается только первым числом знака.

Число -0,03		
1	01111001	11101011100001010001111
^	^	^
Знак	Экспонента	Мантисса

Стандарт также регулирует правила округления чисел, имеется 4 режима округления:

- округление к ближайшему представимому числу

В этом режиме округление происходит к ближайшему точному значению или, в неоднозначной ситуации, к числу с младшим битом мантиссы 0

- округление к $-\infty$;

В этом режиме округление происходит к ближайшему числу не больше точного значения числа (т. к. знак бесконечности -)

- округление к $+\infty$;

В этом режиме округление происходит к ближайшему числу не меньше точного значения числа (т. к. знак бесконечности +)

- округление к 0.

В этом режиме округление происходит к ближайшему числу не больше по абсолютной величине точного значения.

Особые значения. Встречаются числа, обработка которых происходит особым нестандартным образом. Форматы с плавающей запятой предусматривают некоторые значения, которые обрабатываются особым образом. Рассмотрим эти значения:

– Ноль. Очевидно, что значение нуля невозможно написать в нормализованном виде, использующийся для обычных чисел, поэтому для определения значения 0 используется шаблон, состоящий из нулей. При необходимости с помощью разряда знака можно представить значение.

- Бесконечность + и -. Знак в представлении будет равен знаку бесконечности, секция экспоненты заполняется единицами, а вся мантисса нулями.

- Субнормальные (денормализованные) числа. Такими зовутся числа, которые находятся ближе к нулю, чем наименьшее число в нормализованном представлении. Если показатель степени представлен одними нулевыми значениями, а значащая часть не равна нулю, то старший бит значащей части обозначается как 0, а не 1. Точность этих чисел снижается по мере увеличения количества начальных нулей в значащей части.

- Неопределенность. Неопределенность получается в результате ошибки выполнения арифметических операций. В таком числе все двоичные разряды порядка равны единице, а мантисса отлична от нуля.

Типы исключений. Стандартом IEEE 754 определены пять типов исключений:

– *invalid operation* (неверная операция). В стандарте приводится список ситуаций, в которых выставляется исключение *invalid operation*. Например любые операции над полученными неопределенностями, сложение бесконечностей с разными знаками или вычитание бесконечностей с одинаковыми знаками, извлечение квадратного корня из отрицательного числа, умножение 0 на бесконечность, частное нулей или бесконечностей;

– *division by zero* (деление на ноль). Исключение *division by zero* возникает, когда делитель равен нулю. Обязательное условие возникновения исключения — не равное нулю или бесконечности делимое. В случае возникновения исключения *division by zero* результат равен бесконечности;

– *overflow* (переполнение). Исключение *overflow* возникает, когда результат операции превышает границу представляемого числа;

– *underflow* (потеря значимости). Исключение *underflow* возникает, когда результатом операции становится вид субнормального числа;

– *inexact* (потеря точности). Исключение *inexact* возникает тогда, когда результат операции отличается от точного значения.

Стандарт IEEE 754 широко применяется в технике и программировании, однако был создан в то время, когда компьютера обладали оперативной памятью всего лишь 1Мб. Переменные, занимающие 8 байт, с развитием индустрии перестали считаться весомыми по сравнению с теми годами, поэтому со временем такое представление чисел совсем устареет и постепенно лишит эффективности работу систем.

Posit как альтернатива IEEE 754

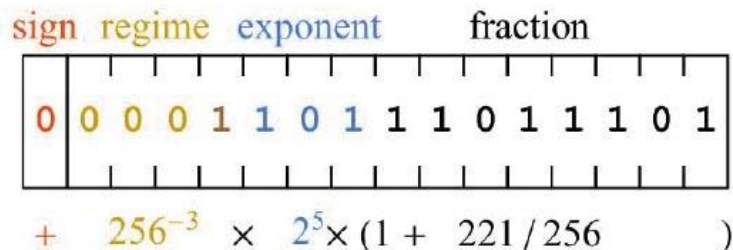
Запись по стандарту IEEE 754 - не единственный способ. Джон Густафсон, известный исследователь и разработчик в сфере компьютерных

вычислений и прикладной математики, со своей командой предложил замену типа float на unum, в последствие доработанный и ставший posit. Формат posit имеет те же составляющие 3 секции знака, экспоненты и дробной части, как и float, но в posit помимо этого обладает дополнительной секцией режима числа. Размеры отделений экспоненты, режима и дробной части не статичны, а варьируются. Как и принято, значение 0 в бите знака соответствует положительному числу, 1 – отрицательному.

Режим числа — это строка из битов, первый бит которой имеет нулевое, либо единичное значение, она продолжается до тех пор, пока не попадет бит противоположной величины, что называют ведущей последовательностью, а количество битов одной величины (с одинаковым значением) длиной этой последовательности. Случается, что режим числа может занимать всё число, в этом случае для остальных секций битов совсем не остается и число с плавающей точкой отображается напрямую без экспоненты и дробной части.

Стандарт posit не требует использования интервальной арифметики или операндов переменного размера, и, как и float, числа posit округляются, если результат не может быть представлен точно. У числа в формате posit есть наглядные преимущества над форматом float, например он имеет больший диапазон и большую точность. Числа posit обеспечивают более высокую точность, чем float того же размера, по словам самого Густафсона, числа формата posit дадут улучшенную производительность и точность при использовании меньшего количества бит.

Рис.2 Альтернатива стандарту IEEE 754, предложенная Густафсоном



За счет того, что для экспоненты, как правило, выделяется меньше битов, чем в стандарте IEEE 754, точность для числа повышается, однако сверхмалые или сверхкрупные числа имеют напротив, меньшую точность.

Задание на практическое занятие:

Представить в стандарте IEEE754 № зачётной книжки.

Контрольные вопросы:

1. В чём разница целочисленной арифметики от арифметики с плавающей точкой?
2. Что такое нормализованные и денормализованные числа ?
3. Кто предложил стандарт IEEE754 ?
4. Что такое мантисса в стандарте IEEE754 ?
5. Что такое экспонента в стандарте IEEE754 ?
6. Какие исключения существуют в стандарте IEEE754 ?
7. Какие особые значения существуют в стандарте IEEE754 ?
8. Является ли стандарт IEEE754 единственно возможным для выполнения вычислений арифметики с плавающей точкой ?

Практическое занятие №3**Метрики параллельных вычислений**

Цель и задачи проведения практического занятия.

Исследование числовых характеристик эффективности распараллеливания вычислительного процесса. Получение практических навыков вычисления метрик параллельных вычислений.

Методические указания по теме. Краткие теоретические сведения.

Метриками параллельных вычислений принято называть некую систему показателей, что предоставляет ряд возможностей для оценки преимуществ системы, которая состоит из n числа процессоров, в плане параллельных вычислений по сравнению с последовательной обработкой информации, осуществляемой только одним процессором.

Базисом для определения метрик являются следующие характеристики вычислений:

n – количество используемых процессоров в системе, осуществляющей параллельные вычисление;

$O(n)$ - объем вычислений. Выражается количеством операций, которые выполняют процессоры системы в процессе решения той или иной задачи;

$T(n)$ - общее время, за которое все процессоры выполняют вычисления. Обычно выражается в секундах.

В системе, состоящей из одного процессора общее время выполнения вычислений равно объёму вычислений, т.е. $O(1) = T(1)$. В общем случае, когда процессоров в системе, выполняющей вычисления более одного, общее время выполнения вычислений должно быть меньше объёма вычислений, т.е. $T(n) < O(n)$. Таким образом, время вычислений можно снизить за счёт разбиения

одной большой задачи на несколько мелких задач, которые будет решать отдельный процессор.

Следует выделять следующие группы метрик:

Первая группа характеризует скорость вычислений. Первая группа метрик параллельных вычислений состоит из двух параметров: индекс параллелизма и ускорение.

Индекс параллелизма (PI – Parallel Index) – это показатель, которым принято обозначать среднюю скорость решения задач, выполняемых параллельно. Выражается в количестве выполняемых операций в единицу времени. Индекс параллелизма рассчитывается по следующей формуле:

$$PI(n) = \frac{O(n)}{T(n)}$$

Ускорение (S – Speedup) – это показатель эффективной скорости выполнения вычислений. Является отношением времени выполнения задачи последовательно на системе из одного процессора к времени выполнения идентичной задачи параллельно на системе из нескольких процессоров. Обычно ускорение должно быть меньше количества процессоров, т.е. ускорение должно удовлетворять условию $S(n) \leq n$. Стоит подчеркнуть тот факт, что и для последовательного, и для параллельного варианта реализации вычислений могут подойти совершенно разные алгоритмы, которые можно называть лучшими. Но когда речь идёт об оценке ускорения, то следует исходить только из наилучших алгоритмов вычислений. Ускорение параллельных вычислений стоит рассчитывать по следующей формуле:

$$S(n) = \frac{T(1)}{T(n)}$$

Вторую группу составляют метрики, которые дают представления об эффективности обеспечения системы дополнительными процессорами для выполнения вычислений.

Эффективность (E – Efficiency) – это показатель, который говорит о целесообразности пополнения количества процессоров за определённую долю ускорения, которую можно достигнуть благодаря параллельным вычислениям, выполняемым одним процессором. Эффективность как правило должна удовлетворять условию, что эффективность должна быть больше отношения единицы к количеству процессоров, но меньше количества процессоров, т.е. $\frac{1}{n} \leq E(n) \leq n$. Более реалистичное получение картины

производительности параллельного выполнения задачи можно получить благодаря моделированию ситуации, в которой алгоритм, выполняемый параллельно, возможно потребует гораздо больше операций, чем алгоритм, выполняемый последовательно. Для расчёта эффективности существует две

формулы: $E(n) = \frac{S(n)}{n}$ и $E(n) = \frac{T(1)}{n * T(n)}$

Утилизация (U – Utilization) – коэффициент, который должен учитывать вклад каждого процессора при параллельном вычислении. Выражается в количестве операций, выполняемых процессором за единицу времени. Коэффициент полезного использования утилизации следует рассчитывать по

двум формулам: $U(n) = R(n) * E(n)$ и $U(n) = \frac{O(n)}{n * T(n)}$ В данном случае учитывается условие, что объём вычислений равен общему времени вычислений, т.е. $O(n)=T(N)$

Третья группа метрик характеризует эффективность параллельных вычислений, выполненных при параллельном и последовательном решении задачи. К третьей группе метрик относятся такие характеристики, как избыточность и сжатие.

Избыточность (R – Redundancy) – это отношение объёмов параллельных вычислений и последовательных вычислений. Данная метрика важна, потому она исходит из абсолютных показателей объёма выполненной вычислительной работ. Также избыточность имеет свойство отражать, насколько аппаратный и программный параллелизм соответствуют друг другу. Избыточность принято считать по следующей формуле: $R(n) = \frac{O(n)}{O(1)}$

Сжатие (C – Compression) – показатель, который математически является метрикой, обратной избыточности. По сути отражает, во сколько объём операций в последовательной однопроцессорной системе больше, чем в параллельной многопроцессорной. Сжатие вычисляют по следующей формуле: $C(n) = \frac{O(1)}{O(n)}$.

В четвертую группу входит такая метрика параллельных вычислений, как качество. Качество (Q – Quality) – это метрика, объединяющая в себе такие 3 параметра из остальных трёх групп, как ускорение, эффективность и сжатие. Именно по этой причине данная метрика считается показателем, который наиболее объективно показывает улучшение производительности системы благодаря применению параллельных вычислений. Для расчёта качества существует две формулы: $Q(n) = S(n) * E(n) * C(n)$ и $Q(n) = \frac{T^3(1)}{nT^2(n)*O(n)}$

Рассмотрим пример применения метрик параллельных вычислений. Предположим, что определённая задача решается системой, состоящей из одного процессора и выполнившая в ходе работы 500 операций, за 500 секунд. Эту же задачу решает другая система, состоящая из 13 процессоров и выполнившая процессе работы 196 операций, за 14 секунд. Необходимо рассчитать по известным формулам показатель повышения производительности благодаря применению параллельных вычислений в решении задачи.

$T(1) = 500$ секунд

$n = 13$ процессоров

$T(13) = 14$ секунд

$O(1) = 500$ секунд

$O(13) = 196$ секунд

1. Рассчитаем индекс параллелизма $PI(n)$:

$$PI(n) = \frac{O(n)}{T(n)} = \frac{196}{14} = 14$$

2. Ускорение $S(n)$:

$$S(n) = \frac{T(1)}{T(n)} = \frac{500}{14} = 35,7$$

3. Эффективность $E(n)$:

$$E(n) = \frac{S(n)}{n} = \frac{35,7}{13} = 2,8$$

4. Утилизация $U(n)$:

$$U(n) = R(n) * E(n) = \frac{O(n)}{n * T(n)} = \frac{196}{13 * 14} = 1,1$$

5. Избыточность $R(n)$:

$$R(n) = \frac{O(n)}{O(1)} = \frac{196}{500} = 0,4$$

6. Сжатие $C(n)$:

$$C(n) = \frac{O(1)}{O(n)} = \frac{500}{196} = 2,6$$

7. Качество Q :

$$Q(n) = S(n) * E(n) * C(n) = 35,7 * 2,8 * 2,6 = 259,9$$

Закон Амдала.

Джин Амдал – один из создателей всемирно знакомой системы IBM 360 в 1967 году представил формулу, отражающую зависимость ускорения вычислений, достигаемого на многопроцессорной ВС, как от количества микропроцессоров, так и от соотношения между поочередной и распараллеливаемой частями программы. Неувязка рассматривалась Амдалом исходя из положения, что размер решаемой задачи с переменной числа микропроцессоров, участвующих в ее разрешении, остается неизменяющимся. Эта постановка свойственна для случаев, когда главным требованием считается скорость вычислений, к примеру, для задач, производимых в реальном времени.

Пусть f – часть операций, которые обязаны производиться поочередно одним из микропроцессоров и $1-f$ – часть, приходящаяся на распараллеливаемую долю программы. Таким образом ускорение, которое имеет возможность быть получено на ВС из n микропроцессоров, по сравнению с однопроцессорным решением не станет превосходить величины:

$$S(n) = \frac{T1}{T(n)} = \frac{1}{\left[f + \frac{1-f}{n} \right]}$$

Закон Амдала демонстрирует, практически что прирост производительности вычислений находится в зависимости от метода задачи и ограничен сверху для ради задачи с $f \neq 0$.

Более того, в случае если принимать во внимание период, необходимый для передачи данных меж связками вычислительной системы, то зависимость периода вычислений от количества узлов станет иметь максимум. Это накладывает лимит на изменчивость ВС, то есть значит, собственно то что с конкретного этапа добавление узлов в систему станет увеличивать период расчета задачи.

Закон Густафсона.

Джон Густафсон из NASA Ames Research пришел к выводу, собственно что наращивание общего объема программы касается в основном распараллеливаемой части. Значит, ускорение возможно увеличить благодаря тому, что, оставаясь буквально постоянной, поочередная доля в общем размере увеличенной программы содержит уже меньший удельный вес.

Пусть f – часть операций, которые обязаны выполняться поочередно одним из микропроцессоров и $1-f$ – часть, приходящаяся на распараллеливаемую долю программы.

Таким образом, для оценки способности ускорения, которое имеет возможность быть получено на ВС из n микропроцессоров, когда размер вычислений возрастает с ростом числа микропроцессоров, применяется выражение, знакомое как закон масштабируемого ускорения или же закон Густафсона.

$$S(n) = f + (1 - f)n = n + (1 - n) \times f$$

Данным образом, закон Густафсона определяет ситуацию, при которой время вычислений с расширением системы не изменяется, но возрастает размер решаемой задачи. Задача такого подхода – за данное время исполнить наибольший размер вычислений.

Закон Густафсона не противоречит закону Амдала. Отличие состоит только в форме утилизации дополненной мощности ВС, образующейся при повышении количества микропроцессоров.

Закон Сана-Ная.

Закон Сана-Ная определяет ускорение в обстановке, когда увеличение размера решаемой задачи ограничено емкостью доступной памяти. В многопроцессорной параллельной ВС любой микропроцессор как правило содержит независимую локальную память относительно маленькой емкости. Общая память ВС появляется объединением локальной памяти всякого микропроцессора ВС. При решении задача делится на подзадачи и делится по множеству микропроцессоров. Подзадача располагается в локальной памяти микропроцессора. Как и в постановке Густафсона, наращивание количества микропроцессоров сопрягается возрастанием объема решаемой задачи, но до предела, обусловленного емкостью доступной памяти. Другими словами, размер задачи возрастает так, дабы любая подзадача всецело занимала локальную память микропроцессора. Эта постановка лежит в базе закона, сформулированного Ксиан-Хе Саном и Лайонелом Наем, и носит название закона ускорения, ограниченного памятью:

$$S(n) = \frac{[f + (1 - f)G(n)]}{[f + (1 - f)\frac{G(n)}{n}]}$$

В предоставленном выражении полагается, собственно что при применении всех n микропроцессоров распараллеливаемая часть задачи может масштабироваться в $G(n)$ раз, f – часть операций, которые обязаны выполняться поочерёдно одним из микропроцессоров и $1-f$ – часть, приходящаяся на распараллеливаемую долю программы.

Выведенное выражение представляет собой обобщение законов Амдала и Густафсона. При $G(n)=1$ величина задачи фиксирована, собственно что соответствует постановке Амдала. Вариант $G(n)=n$ соответствует случаю, когда с наращиванием емкости памяти в n раз рабочая нагрузка ещё растёт в n раз. Это тождественно постановке Густафсона. В случае когда вычислительная нагрузка растёт скорее, чем запросы к памяти ($G(n)>n$), модель с лимитом по памяти выделяет более оптимистичную оценку ускорения.

Задание на практическое занятие:

1. Рассчитать рассмотренные метрики параллельных вычислений для следующих исходных данных:

2. $T(1) = 100000$ секунд
3. $n =$ (число студентов в группе) процессоров
4. $T(n) = 14$ секунд
5. $O(1) = 500$ секунд
6. $O(n) =$ (№ зачётной книжки) секунд

Рассчитать максимальное ускорение по Амдалу, Густаффсону и Сана-Наю для

1. $G(n)=n$
2. $f =$ № по списку в группе

Контрольные вопросы:

1. Для чего применяются метрики параллельных вычислений?
2. Что характеризует индекс параллелизма?
3. Что характеризует ускорение?
4. Что характеризует эффективность?
5. Что характеризует избыточность?
6. Что характеризует утилизация?
7. Что характеризует сжатие?
8. Что характеризует качество?

Практическое занятие №4

Радиаторы для элементов и узлов цифрового компьютера.

Цель и задачи проведения практического занятия.

Ознакомиться с задачами, для решения которых предназначены радиаторы электронных узлов и элементов. Овладеть численным методом расчёта радиатора. Получить практические навыки расчёта радиатора.

Методические указания по теме. Краткие теоретические сведения.

Для всех очевидно, что при работе любой техники происходит нагревание различных частей машины, вычислительная техника – не исключение. И если раньше, с менее мощными вычислительными системами это было не столь критично, то теперь, с развитием и увеличением вычислительных мощностей выделяемого тепла стало достаточно для того, чтобы привести к проблемам. Нагревание может привести к перегреву системы, что может привести (и приведёт) к неисправности машины. Исходя из этого становится очевидной необходимость системы охлаждения цифрового компьютера.

На сегодняшний день существуют 2 способа охлаждения, так называемые активный и пассивный способы. Активный подразумевает принудительный прогон воздуха при помощи дополнительных устройств, вроде вентилятора (Также их называют кулерами). Также есть пассивный способ, который подразумевает естественный, а не вынужденный теплообмен с окружающей средой. Как правило для применения пассивного снижения температуры устанавливают радиаторы, которые и за счёт свойств материала и особенностей конструкции каждого, ускоряют теплообмен между устройством и воздухом.

Радиаторы

Как уже было сказано во вступлении, радиатор – это компонент, ускоряющий передачу тепла от нагревающегося устройства в окружающую среду, то есть в воздух. Для маломощных вычислительных систем, только радиатора будет вполне достаточно для поддержания допустимой температуры устройства.

Как же именно радиатор помогает передавать тепло? Принцип работы радиатора заключается в прямой передаче тепла от нагревающегося компонента (например транзистора) на сам радиатор либо за счёт собственных свойств теплопроводности материала, то есть банально через прямой контакт нагреваемого элемента с «дном» радиатора, либо с помощью тепловых трубок. Радиатор в свою очередь передаёт получаемое тепло в воздух, т.к. будет стремиться к тепловому равновесию (второй закон термодинамики). Для ускорения выхода тепла из радиатора их иногда «чернят».

В настоящий момент радиаторы являются повсеместно применяемым инструментом для охлаждения, даже если не как самостоятельная система, то как элемент охлаждающей системы активного типа.

Эффективность радиатора зависит от нескольких факторов:

1. Площадь теплового соприкосновений с окружающей средой.
2. Конструкционные особенности.
3. Материал, используемый при изготовлении.
4. Скорость и температура проходящего воздуха (чем ниже температура и выше скорость, тем лучше)

Поверхность нагревающегося компонента, как и самого радиатора даже после шлифовки и полировки будут иметь шероховатости, которые не позволят радиатору и нагреваемому элементу достаточно плотно прилегать друг к другу, образуются «воздушные карманы», а воздух, как мы все знаем не является хорошим теплопроводным материалом из-за чего будет значительно снижаться эффективность теплообмена через установленный радиатор, для решения, или по крайней мере смягчения этой проблемы, образовавшиеся воздушные промежутки заполняют теплопроводными пастами.

Охлаждение с одним только радиатором (т.е. пассивное) для процессоров требует применения специальных, довольно крупных радиаторов, которые могут обеспечить высокую скорость отвода тепла даже при низкой скорости проходящего воздушного потока, что будет крайне полезно при сборке тихого компьютера.

Как мы уже было указано, конструкционные особенности являются важным фактором при производстве эффективных радиаторов. В данный момент мы можем наблюдать и приобретать радиаторы 5 основных конструкционных типов.

Прессованные радиаторы.

Наиболее дешевые, общепризнанные и самые распространенные, основной материал, используемый в их производстве — алюминий. По форме представляют из себя мономатериальную конструкцию с пластиной-основанием и рядами тонких листов на ней, которые и формируют поверхность для теплового контакта с воздухом.

Складчатые радиаторы.

Основные материалы, применяемые при производстве — алюминий и медь. Отличается от прессованных радиаторов тем, что на пластине-основании радиатора установлены (чаще всего пайкой, в отличии от прессованных, которые получаются собственно говоря прессованием) не ряды «поверхностей», а загнутая «гармошкой» металлическая лента, которая и исполняет роль поверхности для контакта с воздухом. По сравнению с прессованными радиаторами, данная технология позволяет получать изделия более компактных размеров, но с примерно такой же эффективностью.

Кованые радиаторы.

Для их изготовления используется технология «холодного прессования», которая позволяет придавать поверхности радиатора нужную

форму, в частности кованые радиаторы представляют собой ту же пластину-основание, как и у прежних, но сверху находятся не лента или пластины, а стержни произвольного сечения. Основным материал — алюминий, но зачастую в основание радиатора дополнительно интегрируют медные пластины. Данный конструкционный тип, хоть и дороже в производстве считается менее эффективным.

Составные радиаторы.

Во многом повторяют методику «складчатых» радиаторов, в том смысле, что ребристая поверхность припаивается, но обладают вместе с тем весьма существенным отличием: здесь поверхность формируется уже не лентой-гармошкой, а отдельными тонкими пластинами, закрепленными на пластине-основании радиатора пайкой или стыковой сваркой. Основным используемый материал — медь. Обычно составные радиаторы характеризуются большей эффективностью, чем прессованные и складчатые, но это наблюдается только при условии жесткого контроля качества производственных процессов.

Точёные радиаторы.

На сегодняшний день точёные радиаторы считаются наиболее продвинутыми и эффективными (но при этом и самыми дорогими). Они производятся методом «прецизионной механической обработки монолитных заготовок», это значит, что при изготовлении точётых радиаторов применяются высокоточные станки, которые могут сделать рёбра радиатора максимально тонкими, особенно по отношению к прочим методам создания радиаторов. Основные материалы – алюминий и медь.

Процесс расчёта достаточной площади радиатора

Для рассеивания 1 ватта тепла, выделяемого полупроводниковым прибором, достаточно использовать площадь теплоотвода, равную 30 квадратным сантиметрам.

Существует формула для расчёта:

$$P = (T_2 - T_1) / (Q + Q_1 + Q_2) \text{ где}$$

P - рассеиваемая на мощность.

Q - тепловое сопротивление между радиатором и окружающей средой. Спецификация большинства радиаторов содержит этот параметр.

Q1 - тепловое сопротивление между кристаллом и корпусом силовых элементов обычно приводится в справочнике и обозначается RthJC. Значение этой величины в основном зависит от типа корпуса и у современных транзисторов составляет величину 0,4-1,5 (°C/Вт) или (K/Вт).

Q2 - значение теплового сопротивление корпус-радиатор стремится к нулю в тех случаях, когда мы прикручиваем транзистор к отполированной поверхности радиатора без изолирующих прокладок, или используем тонкие современные подложки из оксида алюминия (Al2O3), нитрида алюминия (AlN), или оксида бериллия (BeO). В случае применения слюды значение теплового сопротивления может составлять 0.2-1.5 (°C/Вт), в зависимости от толщины прокладки.

T_2 - максимальная температура кристалла транзистора, составляет для мощных транзисторов величину 120-175°C.

T_1 - максимально допустимая температура внутри корпуса, в котором находится радиатор, либо максимальная температура окружающей среды, если ребра радиатора выведены наружу.

Расчёт радиатора для транзистора КТ315А ($T_2=120$, $T_1=100$, $Q=130$, $Q_1=1.2$, $Q_2=1.3$)

$$P = (T_2 - T_1) / (Q + Q_1 + Q_2) = 0.15 \Rightarrow \text{требуемая площадь радиатора } 4.5 \text{ см}^2$$

Задание на практическое занятие:

Рассчитать радиатор для транзистора, выбранного преподавателем.

Контрольные вопросы:

1. Для чего предназначен радиатор элемента цифрового компьютера?
2. В чём отличие радиатора от кулера?
3. В чём особенности пресованных радиаторов?
4. В чём особенности складчатых радиаторов?
5. В чём особенности кованых радиаторов?
6. В чём особенности составных радиаторов?
7. В чём особенности точёных радиаторов?

Практическое занятие №5

Кулеры.

Цель и задачи проведения практического занятия.

Ознакомиться с задачами, для решения которых предназначены кулеры электронных узлов и элементов. Овладеть численным методом расчёта кулера. Получить практические навыки расчёта кулера.

Методические указания по теме. Краткие теоретические сведения.

Кулером в компьютере называют устройство, которое представляет собой совокупность вентилятора и радиатора. Самыми лучшими кулерами принято считать те, которые максимально эффективно охлаждают процессор и при этом не издают шум, который может раздражать. В свою очередь шум, который мы слышим при работе компьютера – это шум, работающего вентилятора. Существует некоторая закономерность – больший вентилятор создает меньше шума и обеспечивает лучшее терморегулирование. Если говорить о размерах кулеров, то стоит отметить, что по сей день они постоянно изменяются из-за того, что тепловыделение современных процессоров постоянно растёт. Также вес самих современных процессорных радиаторов

иногда близится к килограмму. Именно поэтому производители стали использовать теплопроводные трубки, которые повышают эффективность охлаждения и к тому же позволяют уменьшить размеры кулера.

Принцип работы кулера.

Если говорить простыми словами о принципе работы кулера, то мы имеем тепло, которое выделяется при работе частей компьютера, оно в свою очередь переходит на радиатор, и в конце вентилятор отводит это тепло. Также обязательно между радиатором и контактной пластиной процессора должен быть слой термопасты, она позволит обеспечить наилучшее соприкосновение и теплопроводность.

К слову о прямом назначении кулера в компьютере, он нужен для охлаждения его частей, таких как: процессор, блок питания и видеокарта, а также вывода горячего воздуха из корпуса. Стоит отметить, что в современных процессорных кулерах блок охлаждения в основном расположен отдельно от подложки, контактирующей с чипом, и соединяются с ней тепловыми трубками.

Теперь разберем более подробно работу кулера. Сам кулер состоит из вентилятора и радиатора. Радиатор имеет большую площадь поверхности, потому что состоит из большого количества металлических пластин. Большой размер радиатора как раз способствует активному рассеиванию тепла. Вентилятор посредством создаваемого мощного потока отводит тепло от радиатора в пространство самого системного блока, где уже встроенные в корпус вентиляторы выводят воздух.

Если говорить об отличиях кулеров в настольных компьютерах и ноутбуках, то выяснится, что устроены кулеры для разных видов компьютеров похожим образом, схожи и принципы работы, но размеры радиатора и вентилятора в ноутбуках меньше, из-за этого в последние устанавливаются процессоры с низким тепловыделением.

Виды конструкций кулеров.

Чаще всего, когда речь идет о видах кулерных устройств, говорят о башенных и классических видах, в самом деле же их больше.

1. Классический. Это самый простой и недорогой вентилятор для персонального компьютера с самым маленьким показателем охлаждения. Также он используется в бюджетных «холодных» системах.

2. Топ-конструкции. Это система охлаждения с тепловыми трубками, которыми соединены радиатор и основание кулера. Обдувающий вентилятор расположен параллельно материнской плате, что позволяет охлаждать не только сам процессор, но и пространство вокруг сокета - конструкции для монтажа процессора. Данная конструкция дает возможность использовать модули памяти с большим радиатором.

3. Башенный тип. Данный вид считается самым производительным вариантом кулера. Этот тип бывает с одной или двумя вертикальными секциями-башнями. Основание и радиатор соединены между собой тепловыми трубками. В данном типе кулера внутренние компоненты компьютера не

нагреваются, из-за того, что вентилятор отводит тепло с радиатора в сторону вентиляторов корпуса. Обычно такая конструкция используется в компьютерах среднего ценового сегмента.

4. С-типа. Трубки данного вида кулеров изогнуты как буква «С». Внизу они запаяны в основании, а сверху соединены с пластинами перпендикулярно материнской плате. Этот вид кулеров похож на башенный, но поток теплого воздуха направляется в сторону материнской платы, поэтому воздушное охлаждение процессора происходит хуже. Но есть и плюс: все соседние элементы обдуваются воздухом от кулера.

5. Комбинированный. Данный вид является достаточно редким, он применяется в дорогих «горячих» моделях. Комбинированный вид конструкции способен обдувать цепи питания на материнской плате. Два радиатора, которые он имеет закреплены на одном основании в горизонтальном и вертикальном положениях.

Проблемы подключения кулера.

Для того, чтобы мы могли без проблемно подключить кулер нужно учесть несколько моментов. Вот они.

Первый момент - подбор места подключения вентилятора зависит от движения потоков воздуха внутри компьютера. Как правило, движение воздуха осуществляется вверх, поэтому используем стандартную инструкцию:

- Вентиляторы на наддув помещаются на боковой, передней и нижней стенках корпуса;
- Модели на выдув – на задней и верхней стенках.

Второй момент – измеряем размер вентилятора и посадочное место под него. Для того, чтобы непосредственно измерить посадочные места замеряется расстояние между центрами крепежных отверстий.

Третий момент – поговорим о самом креплении кулера, тут важно учесть, что типы крепления кулера на процессоре бывают разными. В зависимости от разъема подключение происходит либо к блоку питания посредством разъема Molex, либо к материнской плате посредством разъемов 3-pin и 4-pin. Именно от способа подключения зависит, сможет ли пользователь управлять скоростью вращения вентиляторов с помощью программ, при использовании 4-х контактных проводов мы имеем возможность управлять скоростью.

Также крепление кулера происходит тремя способами: винтовым, креплением «кроватька» и на защелках. Винтовой способ и на защелках подходят для процессора Intel, а крепление «кроватька» только для процессора AMD.

Методика выбора кулера.

В основном выбор кулера сводится к тому, что для начала нужно определить допустимую сумму, которую мы можем на него потратить. При выборе кулера для процессора нужно обращать внимание на его возможности качественно охлаждать, к тому же современные процессоры требуют очень мощного охлаждения.

Сама стоимость кулера для процессора прежде всего зависит от того, из какого материала он изготовлен. Более дорогими являются кулеры, у которых радиатор имеет медную основу, самыми дорогими – кулеры, у которых радиаторы полностью медные.

Также цена кулера для процессора зависит и от ряда свойств: качества его вентилятора, количества подшипников и уровня шума, который он создает при работе. Если говорить об уровне шума, то в среднем значение колеблется от 15 до 25 Дб. И если при 15 Дб практически не заметно жужжание кулера, то при 25-30 Дб шум будет уже ощутимым.

Пример расчёта кулера.

Вычисление потока воздуха для процессора
Процессор AMD Ryzen 5 4500U with Radeon Graphics
Эффективность радиатора определяется тепловым сопротивлением по отношению к поверхности кристалла:

$$R = \frac{(T-t)}{P}$$

R – тепловое сопротивление

T = 60°C – температура процессора, которую пользователь хочет видеть при максимальной нагрузке

t = 25°C – температура воздуха

P₀ = 15 Вт – тепловая мощность процессора (TDP, указывается в характеристиках).

$$R = \frac{(60-25)}{15} = 2,3$$

Для того, чтобы отводить тепло от процессора, требуется поток воздуха. Именно CFM определяет, какой объем воздуха будет проходить через вентилятор. Для расчета воздушного потока CFM (cubic feet per minute) – Объем проходимого воздуха через вентилятор в минуту. Измеряется в кубических футах за минуту. 1 куб. м = 35 CFM.) воспользуемся следующей формулой:

$$CFM = \frac{P_1}{990 * 1,2 * P_0 * R + 0,00047}$$

$P_1 = S * 5,7 * 10^{-8} * T^4$ – кол-во рассеиваемого тепла телом с температурой T и площадью поверхности S (закон Стефана-Больцмана).

В данном случае температура $T = T_0 + P_0 * R$, где P₀ – тепловыделение процессора, R – тепловое сопротивление процессора, T₀ – температура воздуха внутри корпуса (обычно для ATX- корпусов эта величина составляет 300-305 градусов Кельвина, или примерно 30 градусов Цельсия)

$$P_1 = 8464 * 5,7 * 10^{-8} * 64,5^4 = 8350,06$$

$$CFM = \frac{8350,06}{990 * 1,2 * 15 * 2,3 + 0,00047} = 34,8$$

Задание на практическое занятие:

Рассчитать кулер для процессора, выбранного преподавателем.

Контрольные вопросы:

1. Для чего предназначен кулер процессора цифрового компьютера?
2. Почему для процессора недостаточно радиатора?
3. В чём особенности классических кулеров?
4. В чём особенности кулеров топ-конструкций?
5. В чём особенности кулеров башенного типа?
6. В чём особенности кулеров С-типа?
7. В чём особенности комбинированных кулеров?
8. Что показывает коэффициент CRM?

Практическое занятие №6

Надёжность вычислительных систем

Цель и задачи проведения практического занятия.

Ознакомиться с подходами к оценке надёжности и работоспособности вычислительных систем. Ознакомиться с основными метриками, характеризующими надёжность вычислительной системы. Овладеть методиками получения количественных оценок выбранных метрик надёжности. Получить практические навыки вычисления оценок показателей надёжности.

Методические указания по теме. Краткие теоретические сведения.

Функция надёжности.

Функция надёжности (вероятность безотказной работы) – самый известный и распространённый из показателей надёжности вычислительной системы. Она сообщает сведения о производительности машины на определенном отрезке времени или, иными словами, описывает умение машины предоставить на отрезке времени потенциально возможную производительность. Функция надёжности вычислительной машины

$$r(t) = P\{\forall \tau \in [0, t) \rightarrow \omega(\tau) = 1\}.$$

где $P\{\forall \tau \in [0, t) \rightarrow \omega(\tau) = 1\}$ – возможность того, что для любого τ , относящегося к отрезку времени $[0, t)$, производительность $\omega(\tau)$ ЭВМ соответствует 1, т.е. равняется потенциально возможной.

Для подтверждения свойств функции $r(t)$ и установления показателей надёжности, образованных от $r(t)$, лучше дать такое определение:

$r(t) = P\{\xi > t\}$, заключающегося в том, что момент ξ появления 1 отказа при работе вычислительной машины в назначенных условиях использования последует после времени $t \geq 1$.

$r(t)$ имеет следующие свойства:

1) $r(0) = 1$; действительно, т. к. событие $\xi > 0$ является истинным, то $P\{\xi > 0\} = 1$;

2) $r(+\infty) = 0$; т. к. событие $\xi > +\infty$ является недопустимым (или, иначе, событие, заключающееся в том, что машина способна осуществлять заданную функцию на конечном отрезке времени, является истинным), то, следует, что $P\{\xi > +\infty\} = 0$;

3) $r(t_1) \geq r(t_2)$ для $t_1 \leq t_2$; учитывая, что события $\xi > t_2$ и $t_2 \geq \xi > t_1$ считаются не совместными, поэтому по теореме сложения вероятностей имеем

$$\begin{aligned} r(t_1) &= P\{\xi > t_1\} = P\{(\xi > t_2) \cup (t_2 \geq \xi > t_1)\} = \\ &= P\{\xi > t_2\} + P\{t_2 \geq \xi > t_1\} \geq P\{\xi > t_2\} = r(t_2) \end{aligned}$$

Функция надежности в стационарном режиме: $r(t) = \exp(-\lambda t)$

Функция готовности.

Функция готовности ЭВМ $s(i, t) = P_1(i, t) = P\{i; \omega(t) = 1\}$,

$\{i; \omega(t) = 1\}$, есть возможность того, что машина будет иметь в момент времени $t \geq 0$ производительность, равную 1, то есть она (производительность) равна потенциально возможной.

Функция готовности вычислительной машины имеет следующие свойства:

1) $s(0, 0) = 0, s(1, 0) = 1$;

2) $s(i, +\infty) = s = \text{const}$, $0 < s < 1, i \in E_0^1$;

3) $s(0, t) \leq s(0, t_2), s(1, t_1) \geq s(1, t_2)$ для $t_1 \leq t_2$

Функция готовности в переходном режиме функционирования:

$$s(0, t) = \frac{\mu}{\lambda + w\mu} - \frac{\mu}{\lambda + \mu} * e^{-(\lambda + \mu)t}$$

$$s(1, t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} * e^{-(\lambda + \mu)t}$$

Готовность в стационарном режиме:

$$s = \lim_{t \rightarrow \infty} s(i, t) = \frac{\mu}{\lambda + \mu}$$

Функция осуществимости решения задач

Цель функционирования ЭВМ – решение поступивших задач (выполнение программ). Однако введенные показатели надёжности ЭВМ устанавливают взаимосвязь лишь между потенциально возможной производительностью и надёжностью (безотказностью, готовностью)

ремонтпригодностью) машины, т.е. характеризуют качество функционирования ЭВМ безотносительно к процессу решения задач. Этот пробел можно устранить, если использовать для характеристики работы ЭВМ функцию осуществимости решения задач:

$$f(t) = r(t)\varphi(t),$$

где $r(t)$ – вероятность безотказной работы ЭВМ; $\varphi(t) = P\{0 \leq \eta \leq t\}$, т.е. $\varphi(t)$ – это вероятность события $\{0 \leq \eta \leq t\}$; η – случайная величина, являющаяся моментом решения задачи на работоспособной (абсолютно надёжной) ЭВМ. В качестве закона распределения времени решения задач на ЭВМ может быть взят экспоненциальный:

$$\varphi(t) = 1 - e^{-\beta t}, \text{ где } \beta - \text{интенсивность решения задач на машине.}$$

Говорят, что решение задачи на ЭВМ осуществимо, если для некоторого t одновременно выполняются $f(t) \geq f^0$ и $t \leq t^0$, где f^0 и t^0 – «пороги осуществимости»; они выбираются из практических соображений. Интерес представляет также величина $f(t_m) = \max f(t)$, которая отыскивается численными методами.

Известно, что в нормальных условиях эксплуатации ЭВМ интенсивность отказа $\lambda = \text{const}$, а функция надёжности равна: $r(t) = e^{-\lambda t}$;

Тогда функцию осуществимости решения задач $f(t) = r(t)\varphi(t)$ будем рассчитывать по формуле: $f(t) = e^{-\lambda t}(1 - e^{-\beta t})$.

Задание на практическое занятие:

Рассчитать и построить графики функций надёжности, готовности и осуществимости цифрового компьютера в стационарном режиме работы для следующих показателей:

$$\begin{aligned} \mu &= 10 \\ \lambda &= (10^{-\mu}) \text{ по списку} \\ \beta &= 0.001 \end{aligned}$$

Контрольные вопросы:

1. Что описывает функция надёжности?
2. Что описывает функция готовности?
3. Что описывает функция осуществимости?
4. В чём отличия переходного и стационарного режимов с точки зрения надёжности?
5. Как связаны функция надёжности и вероятность безотказной работы?
6. Что показывает интенсивность решения задач?
7. Что показывает интенсивность отказов?
8. Что показывает интенсивность восстановлений?