Всероссийский Заочный Финансово – Экономический Институт

Факультет: финансово – кредитный

Специальность: финансы и кредит группа вечерняя

Курсовая работа

по дисциплине «Информатика»

на тему «Назначение, классификация и эволюция операционных систем»

Москва – 2008г.

## Оглавление

Теоретическая часть 3

Введение 3

Назначение операционных систем 4

Перечислим основные функции операционных систем 8

Загрузка в оперативную память подлежащих исполнению программ 8

Обслуживание всех операций ввода-вывода 8

Эволюция и классификация ОС 10

Заключение 18

Практическая часть 20

I. Общая характеристика задачи 20

II. Формирование ведомости зарплаты 25

III. Данные результатной таблицы 29

Список использованной литературы 33

## Теоретическая часть

## Введение

Операционная система (ОС) является основой системного ПО, под управлением которыми осуществляется начальная загрузка компьютера, управление работой всех его устройств и проверка их работоспособности, управление файловой системой компьютера, загрузка пользовательских приложений и распределение ресурсов компьютера между ними, поддержка пользовательского интерфейса и др. К числу широко известных семейств операционных систем относятся DOS, WINDOWS, UNIX, NETWARE и др.

Операционная система (ОС) представляет собой совокупность программ, выполняющих две функции: предоставление пользователю удобств виртуальной машины[[1]](#footnote-1) и повышение эффективности использования компьютера при рациональном управлении его ресурсами.

Процессор компьютера выполняет команды, заданные на машинном языке. Непосредственная подготовка таких команд требует от пользователя знаний языка и специфики построения и взаимодействия аппаратных средств. Так, например, для доступа к хранящейся на магнитном носителе информации необходимо указать номера блоков на диске и номера секторов на дорожке, определить состояние двигателя механизма перемещения головок записи/считывания, обнаружить наличие и типы ошибок, выполнить их анализ и пр. Требовать этих знаний от всех пользователей практически невозможно. Поэтому и возникла необходимость в создании ОС – совокупности программ, скрывающих от пользователя особенности физического расположения информации и выполняющих обработку прерываний[[2]](#footnote-2), управление таймерами и оперативной памятью. В результате пользователю предоставляется виртуальная машина, реализующая работу на логическом уровне.

## Назначение операционных систем

Операционные системы относятся к системному программному обеспечению. Все программное обеспечение разделяется на системное и прикладное. К системному программному обеспечению принято относить такие программы и комплексы программ, которые являются общими, без которых невозможно выполнение или создание других программ. История появления и развития системного программного обеспечения началась с того момента, когда люди осознали, что любая программа требует операций ввода-вывода данных. Это произошло в далекие 50-е годы прошлого столетия. Собственно операционные системы появились чуть позже.

Действительно, если мы не будем иметь возможности изменять исходные данные и получать результаты вычислений, то зачем вообще эти вычисления? Очевидно, что исходные данные могут вводиться различными способами. На практике используются самые разнообразные устройства и методы. Например, мы можем вводить исходные значения с клавиатуры, задавать нужные действия или функции с помощью указателя мыши, считывать записи из файла, снимать оцифрованные значения с датчиков и т.д. Часть исходных данных может быть передана в программу через область памяти, в которую предварительно другая программа занесла свои результаты вычислений. Способов много. Главное – выполнить в программе некоторые действия, связанные с получением исходных данных.

Аналогично, и вывод результатов может быть организован, например, на соответствующие устройства и в форме, удобной для восприятия ее человеком. Либо результаты расчетов будут отправляться программой на какие-нибудь исполнительные устройства, которые управляются компьютером. Наконец, мы можем организовать запись полученных значений на некие устройства хранения данных (с целью их дальнейшей обработки).

Программирование операций ввода-вывода относится к одной из самых трудоемких областей создания программного обеспечения. Здесь речь идет не об использовании операторов типа READ или WRITE в языках высокого уровня. Речь идет о необходимости создать подпрограмму в машинном виде, уже готовую к выполнению на компьютере, а не написанную с помощью некоторой системы программирования (систем программирования тогда еще не было), подпрограмму, вместо обычных вычислений управляющую тем устройством, которое должно участвовать в операциях ввода исходных данных или выводов результатов. При наличии такой подпрограммы программист может обращаться к ней столько раз, сколько операций ввода-вывода с этим устройством ему требуется. Для выполнения этой работы программисту недостаточно хорошо знать архитектуру вычислительного комплекса и уметь создавать программы на языке ассемблера. Он должен отлично знать и интерфейс, с помощью которого устройство подключено к центральной части компьютера, и алгоритм функционирования устройства управления устройства ввода-вывода.

Очевидно, что имело смысл создать набор подпрограмм управления операциями ввода-вывода и использовать его в своих программах, чтобы не заставлять программистов каждый раз заново программировать все эти операции. С этого и началась история системного программного обеспечения. Впоследствии набор подпрограмм ввода-вывода стали организовывать в виде специальной библиотеки ввода-вывода, а затем появились и сами операционные системы. Основной причиной их появления было желание автоматизировать процесс подготовки вычислительного комплекса к выполнению программы.

В 50-е годы взаимодействие пользователей с вычислительным комплексом было совершенно иным, чем ныне. Программист-кодер (от англ. coder – кодировщик) – специально подготовленный специалист, знающий архитектуру компьютера и язык(и) программирования, - по заказу составлял текст программы, часто по уже готовому алгоритму, разработанному программистом-алгоритмистом. Текст этой программы затем отдавался оператору, который набирал его на специальных устройствах и переносил на соответствующие носители. Чаще всего в качестве носителей использовались перфокарты или перфолента. Далее колода с перфокартами передавалась в вычислительный зал, где для вычислений по этой программе требовать следующие действия:

1. Оператор вычислительного комплекса с пульта вводил в рабочие регистры центрального процессора и в оперативную память компьютера ту первоначальную программу, которая позволяла считать в память программу для трансляции исходных кодов и получения машинной (двоичной) программы (проще говоря, транслятор, который тоже хранился на перфокартах или перфоленте).

2. Транслятор считывал исходную программу, осуществлял лексический разбор исходного текста, и промежуточные результаты процесса трансляции зачастую так же выводили на перфокарты (перфоленту). Трансляция – сложный процесс, часто требующий нескольких проходов. Порой для выполнения очередного прохода приходилось в память компьютера загружать с перфокарт и следующую часть транслятора, и промежуточные результаты трансляции. Ведь результат трансляции выводился также на носители информации, поскольку объем оперативной памяти был небольшим, а задача трансляции – это очень сложная задача.

3. Оператор загружал в оперативную память компьютера полученные двоичные коды, оттранслированной программы и подгружал двоичные коды тех системных подпрограмм, которые реализовывали управлениями операциями ввода-вывода. После этого готовая программа, расположенная в памяти, могла сама считывать исходные данные и осуществлять необходимые вычисления. В случае обнаружения ошибок на одном из этих этапов или после анализа полученных результатов весь цикл необходимо было повторить.

Для автоматизации труда программиста (кодера) стали разрабатывать специальные алгоритмические языки высокого уровня, а для автоматизации труда оператора вычислительного комплекса была разработана специальная управляющая программа, загрузив которую в память один раз оператор мог ее далее использовать неоднократно и более не обращаться к процедуре программирования ЭВМ через пульт оператора. Именно эту управляющую программу и стали называть операционной системой. Со временем на нее стали возлагать все больше и больше задач, она стала расти в объеме. Прежде всего разработчики стремились к тому, чтобы операционная система как можно более эффективно распределяла вычислительные ресурсы компьютера, ведь в 60-е годы операционные системы уже позволяли организовать параллельное выполнение нескольких программ. Помимо задач распределения ресурсов появились задачи обеспечения надежности вычислений. К началу 70-х годов диалоговый режим работы с компьютером стал преобладающим, и у операционных систем стремительно начали развиваться интерфейсные возможности. Термин интерфейс (interface) обозначает целый комплекс спецификаций, определяющих конкретный способ взаимодействия пользователя с компьютером.

На сегодняшний день можно констатировать, что операционная система (ОС) представляет собой комплекс системных управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между аппаратурой компьютера и пользователем с его задачами, а с другой стороны, предназначены для наиболее эффективного расходования ресурсов вычислительной системы и организации надежных вычислений.

## Перечислим основные функции операционных систем

Прием пользователя (или оператора системы) заданий, или команд, сформулированных на соответствующем языке, и их обработка. Задания могут передаваться в виде текстовых директив (команд) оператора или в форме указаний, выполняемых с помощью манипулятора (например, с помощью мыши). Эти команды связаны с запуском (приостановкой, остановкой) программ, с операциями над файлами (получить перечень файлов в текущем каталоге, создать, переименовать, скопировать, переместить тот или иной файл и др.), хотя имеются и иные команды.

## Загрузка в оперативную память подлежащих исполнению программ

Распределение памяти, а в большинстве современных систем и организация виртуальной памяти.

Запуск программы (передача ей управления, в результате чего процессор исполняет программу).

Идентификация всех программ и данных.

Прием и использование различных запросов от выполняющихся приложений. Операционная система умеет выполнять очень большое количество системных функций (сервисов), которые могут быть запрошены из выполняющейся программы. Обращение к этим сервисам осуществляется по соответствующим правилам, которые и определяют интерфейс прикладного программирования (Application Program Interface, API) этой операционной системы.

## Обслуживание всех операций ввода-вывода

Обеспечение работы систем управлений файлами (СУФ) и/или систем управления базами данных (СУБД), что позволяет резко увеличить эффективность всего программного обеспечения.

Обеспечение режима мультипрограммирования, то есть организация параллельного выполнения двух или более программ на одном процессоре, создающая видимость их одновременного исполнения.

Планирование и диспетчеризация задач в соответствии с заданными между выполняющимися программами.

Для сетевых операционных систем характерной является функция обеспечения взаимодействия связанных между собой компьютеров.

Организация механизмов обмена сообщениями и данными между выполняющимися программами.

Защита одной программы от влияния другой, обеспечение сохранности данных, защита самой операционной системы от исполняющихся на компьютере приложений.

Аутентификация и авторизация пользователей (для большинства диалоговых операционных систем). Под аутентификацией понимается процедура проверки имени пользователя и его пароля на соответствие тем значениям, которые хранятся в его учетной записи[[3]](#footnote-3). Очевидно, что если входное имя (login[[4]](#footnote-4)) пользователя и его пароль совпадают, то, скорее всего, это и будет тот самый пользователь. Термин авторизация означает, что в соответствии с учетной записью пользователя, который прошел аутентификацию, ему (и всем запросам, которые будут идти к операционной системе от его имени) назначаются определенные права (привилегии), определяющие, что он может делать на компьютере.

Удовлетворение жестким ограничениям на время ответа в режиме реального времени (характерно для ОС реального времени).

Обеспечение работы систем программирования, с помощью которых пользователи готовят свои программы.

Предоставление услуг на случай частичного сбоя системы.

Операционная система изолирует аппаратное обеспечение компьютера от прикладных программ пользователей. И пользователь, и его программы взаимодействуют с компьютером через интерфейсы операционной системы.

## Эволюция и классификация ОС

Эволюция ОС во многом обусловлена совершенствование аппаратной базы ЭВМ.

Программирование ламповых вычислительных устройств, ориентированных на решение специализированных прикладных задач, выполнялось на машинном языке (языке программирования, представляющем программу в форме, позволяющей непосредственно выполнять ее техническими средствами обработки данных). Организация вычислительного процесса в этом случае осуществлялась обслуживающим персоналом вручную с пульта управления. ОС для этих ЭВМ практически отсутствовали.

Компьютеры, построенные на полупроводниковых элементах, стали более компактными, надежными и применялись при решении более широкого класса прикладных задач. Появились первые алгоритмические языки, компиляторы (компиляторы – программы, используемые для компиляции – перевода написанной на алгоритмическом языке программы на язык, близкий к машинному) и системы пакетной обработки. Эти системы явились прообразом современных ОС. Основное их назначение – увеличение загрузки процессора.

Переход от отдельных полупроводниковых элементов типа транзисторов к интегральным микросхемам сопровождался созданием семейств программно-совместимых машин, например, семейства IBM/360, EC ЭВМ. ОС этих компьютеров ориентировались на обслуживание вычислительных систем с разнообразными периферийными устройствами и в различных областях деятельности. Особенностью таких ОС стало мультипрограммирование – способ организации вычислительного процесса, при котором на одном процессоре попеременно выполняются несколько приложений. Например, пока одно приложение осуществляет операции ввода-вывода, процессор выполняет вычислительные операции другого. Образовался новый тип ОС – системы разделения времени, которая позволяет создать для каждого пользователя иллюзию единоличной работы с компьютером. Появление больших (БИС) и сверхбольших интегральных схем (СБИС) обеспечило широкое распространение компьютеров и их использование неспециалистами в области программирования. Это потребовало разработки дружественного, интуитивно понятного программного интерфейса. Развитие средств коммуникаций обусловило развитие сетевых ОС.

К современным ОС предъявляются следующие требования:

совместимости – ОС должна включать средства для выполнения приложений, подготовленных для других ОС;

переносимости – обеспечение возможности переноса ОС с одной аппаратурной платформы на другую;

надежности и отказоустойчивости – предполагает защиту ОС от внутренних и внешних ошибок, сбоев и отказов;

безопасности – ОС должна содержать средства защиты ресурсов одних пользователей от других;

расширяемости – ОС должна обеспечивать удобства внесения последующих изменений и дополнений;

производительности – система должна обладать достаточным быстродействием.

Классификация ОС. Обычно общение пользователя с машиной протекает в интерактивном режиме. При этом темп решения задачи определяется реакцией пользователя. Если принять время реакции пользователя постоянным, то можно считать, что сложность задачи ограничивается быстродействием технических средств (при необходимости оно повышается экстенсивными методами: применением более быстродействующей элементной базы, использованием многопроцессорных или многомашинных систем). Но это односторонний подход. Огромные возможности таятся в организационных мероприятиях, к которым относится выбор оптимальных режима работы и дисциплин обслуживания; и то и другое реализуется операционными системами.

Режимы работы ПЭВМ в первую очередь определяются количеством задач, параллельно решаемых на машине (реализуемых программ). По этому признаку ОС разделяются на многозадачные и однозадачные, поддерживающие и не поддерживающие многонитевую обработку, многопользовательские и однопользовательские, на многопроцессорные и однопроцессорные.

По числу одновременно выполняемых задач выделяют ОС:

однозадачные ОС (MS-DOS, ранние версии PS DOS);

многозадачные (OS/2, UNIX, Windows).

Однозадачные ОС предоставляют пользователю виртуальной машины и включают средствами управления файлами, периферийными устройствами и средства общения с пользователем. Многозадачные ОС дополнительно управляют разделением между задачами совместно используемых ресурсов. Среди вариантов реализации многозадачности выделяют две группы алгоритмов распределения процессорного времени:

невытесняющая многозадачность (NetWare, Windows 3. x и 9. х);

вытесняющая многозадачность (Windows NT, OS/2, UNIX).

В первом случае активный процесс по окончании сам передает управление ОС для выбора из очереди другого процесса. Во втором – решение о переключении процесса с одного процесса с одного процесса на другой принимает не активный процесс, а ОС.

Поддержка многонитевости предполагает возможность выполнения некоторых команд программы практически в один и тот же момент. Многонитевая ОС разделяет процессорное время не между задачами, а между отдельными ветвями (нитями) алгоритмов их решения (многозадачность внутри одной задачи).

По числу одновременно работающих пользователей выделяют ОС:

однопользовательские (MS-DOS, Windows 3. x, ранние версии OS/2);

многопользовательские (UNIX, Windows NT).

Отличием многопользовательских систем является наличие средств защиты информации пользователей от несанкционированного доступа.

Многопроцессорная обработка предполагает поддержку работы нескольких процессоров и приветствует в ОС Solaris 2. x фирмы Sun, OS/2 фирмы IBM, Windows NT фирмы Microsoft, NetWare 4.1. фирмы Novell и др.

Многопроцессорные ОС делятся на асимметричные и симметричные. Асимметричная ОС выполняется на одном из процессоров системы, распределяя прикладные задачи по остальным процессорам. Симметричные ОС децентрализована и использует все процессоры, разделяя между ними системные и прикладные задачи.

Очевидно, что ОС персонального компьютера менее сложна, чем ОС мэйнфреймов и суперЭВМ. Отличием сетевой ОС от локальной являются средства передачи данных между компьютерами по линиям связи и реализации протоколов передачи данных, например IP, IPX и др.

Кроме ОС, ориентированных на определенный тип аппаратной платформы, существуют мобильные ОС, легко переносимые на разные типы компьютеров. В таких ОС (например, UNIX) аппаратно-зависимые места локализованы и при переносе системы на новую платформу переписываются. Аппаратно-независимая часть реализуется на языке программирования высокого уровня на языке С, и перекомпилируется при переходе на другую платформу.

Динамическое перераспределение ресурсов ПЭВМ между несколькими задачами (способ коллективного пользования) обеспечивает либо сокращение суммарного времени решения нескольких задач, либо уменьшение времени начала обработки задачи. И то и другое при одном «обслуживающем приборе», например процессоре или принтере, достигается за счет увеличения времени решения одной (каждой) задачи. Порядок прохождения задач в машине определяет режимы реализации способа коллективного пользования: пакетной обработки и разделения времени. Пакетная обработка предполагает решение нескольких задач примерно в том же режиме, в каком работает преподаватель во время устного экзамена – его внимание занимает один из студентов. По завершении опроса внимание переключается на следующего (полагаем, что обслуживание студентов и есть решение задачи).

В режиме разделения времени процесс выполнения программ разбивается на циклы. Внутри каждого цикла обязательно появляется (если она еще не решена) та базовая задача, для которой выделен данный цикл. Остальные задачи могут решаться в этом же цикле, если их решение не препятствует решению базовой. Соседние циклы выделяются для решения разных задач, поэтому существенно увеличивается время решения каждой из них. Вместе с тем на ожидание выполнения очередной части задачи затрачивается не более чем N-1 циклов (N – число разноименных циклов, выделенных для решения N различных задач). При соответствующем выборе длительности цикла у пользователя (из-за его инерционности) создается иллюзия работы в реальном масштабе времени. Работой в реальном масштабе времени называется режим работы, при котором существуют предельные ограничения на время решения задачи, накладываемые внешней средой. Для системы управления это означает, что обработка информации должна протекать со скоростью, превышающей скорость реально протекающего управляемого процесса, с тем, чтобы имелся определенный запас времени для принятия решений и формирования соответствующих управляющих воздействий.

Для реальных условий работы характерно то, что моменты поступления задач на обработку определяются не темпом работы обслуживающего прибора (ПЭВМ, принтера и пр), а процессами, протекающими вне его. В соответствии с этим обслуживающий прибор должен решать определенную совокупность задач (реакция на сбои оборудования, заявка на связь в сети, принудительное создание резервной копии и т.д.).

При конечном быстродействии обслуживающего прибора поступающие заявки не могут быть выполнены сразу, а становятся в очередь. Процесс выбора заявки из множества ожидающих обслуживания называется диспетчеризацией, а правило диспетчеризации – дисциплиной обслуживания. Дисциплин обслуживания много, например «в порядке поступления» (FIFO – First Input First Output), «в обратном порядке» (LIFO – Last Input First Output) и др. Для сокращения времени ожидания (времени пребывания в очереди) отдельным заявкам предоставляется преимущественное право на обслуживание, называемое приоритетом, который характеризуется целым положительным числом. Наивысший приоритет назначается ОС.

Итак, при создании двоичных машинных программ прикладные программисты могут вообще не знать многих деталей управления конкретными ресурсами вычислительной системы, а должны только обращаться к некоторой программной подсистеме с соответствующими вызовами и получать от нее необходимые функции и сервисы. Эта программная подсистема и есть операционная система, а набор ее функций и сервисов, а также правила обращения к ним как раз и образуют то базовое понятие, которое мы называем операционной средой. Можно сказать, что термин «операционная среда» означает соответствующие интерфейсы, необходимые программам и пользователям для обращения к управляющей (супервизорной) части операционной системы с целью получить определенные сервисы.

Системных функций бывает много, они определяют те возможности, которые операционная система предоставляет выполняющимся под ее управлением приложениям. Такого рода системные запросы (вызовы системных операций, или функций) либо явно прописываются в тексте программы программистами, либо подставляются автоматически самой системой программирования на этапе трансляции исходного текста разрабатываемой программы. Каждая операционная система имеет свое множество системных функций; они вызываются соответствующим образом, по принятым в системе правилам. Совокупность системных вызовов и правил, по которым их следует использовать, как раз и определяет интерфейс прикладного программирования (API). Очевидно, что программа, созданная для работы в некоторой операционной системе, скорее всего не будут работать в другой операционной системе, поскольку API у этих операционных систем различаются. Стараясь преодолеть это ограничение, разработчики операционных систем стали создавать так называемое программные среды. Программную (системную) среду следует понимать как некоторое системное программное окружение, позволяющее выполнить все системные запросы от прикладной программы. Та системная программная среда, которая непосредственно образуется кодом операционной системы, называется основной, естественной, или нативной (native). Помимо основной операционной среды в операционной системе могут быть организованы (путем эмуляции иной операционной среды) дополнительные программные среды. Если в операционной системе организована работа с различными операционными средами, то в такой системе можно выполнять программы, созданные не только для данной, но и для других операционных систем. Можно сказать, что программы создаются для работы в некоторой заданной операционной среде. Например, можно создать программу для работы в среде DOS. Если такая программа все функции, связанные с операциями с операциями ввода-вывода и с запросами памяти, выполняет не сама, а за счет обращения к системным функциям DOS, то она будет (в абсолютном большинстве случаев) успешно выполняться и в MS DOS, и в PS DOS, и в Windows 9x, и в Windows 2000, и в OS/2, и даже в Linux.

Итак, параллельное существование терминов «операционная система» и «операционная среда» вызвано тем, что операционная система может поддерживать несколько операционных сред. Почти все современные 32-разрядные операционные системы, созданные для персональных компьютеров, поддерживают по нескольку операционных сред. Так операционная система OS/2 Warp, которая в свое время была одной из лучших в этом отношении, может выполнять следующие программы:

основные программы, созданные с учетом соответствующего «родного» 32-разрядного программного интерфейса этой операционной системы;

16-разрядные программы, созданные для систем OS/2 первого поколения;

16-разрядные приложения, разработанные для выполнения в операционной системе MS DOS или PS DOS;

16-разрядные приложения, созданные для операционной среды Windows 3. x;

саму оперативную оболочку Windows 3. x и уже в ней – созданные для нее программы.

А операционная система Windows XP позволяет выполнять помимо основных приложений, созданных с использованием Win32API, 16-разрядные приложения для Windows 3. x, 16-разрядные DOS-приложения, 16-разрядные приложения для первой версии OS/2.

Операционная среда может включать несколько интерфейсов: пользовательские и программные. Если говорить о пользовательских, то например, система Linux имеет для пользователя как интерфейсы командной строки (можно использовать различные «оболочки» - shell), наподобие Norton Commander, например X-Window с различными менеджерами окон - KDE, Gnome и др. Если же говорить о программных интерфейсах, то в тех же операционных системах с общим названием Linux программы могут обращаться как к операционной системе за соответствующими сервисами и функциями, так и к графической подсистеме (если она используется). С точки зрения архитектуры процессора (и персонального компьютера в целом) двоичная программа, созданная для работы в среде Linux, использует те же команды и форматы данных, что и программа, созданная для работы в среде Windows NT. Однако в первом случае мы имеем обращение к одной операционной среде, а во втором – к другой. И программа, созданная непосредственно для Windows, не будет выполняться в Linux; если в ОС Linux организовать полноценную операционную среду Windows, то наша Windows-программа может быть выполнена. В общем, операционная среда – это то системное программное окружение, в котором могут выполняться программы, созданные по правилам работы этой среды.

## Заключение

Итак, операционная система выполняет функции управления вычислениями в компьютере, распределяет ресурсы вычислительной системы между различными вычислительными процессами и образует ту программную среду, в которой выполняются прикладные программы пользователей. Такая среда называется операционной. Последнее следует понимать в том плане, что при запуске программы она будет обращаться к операционной системе с соответствующими запросами на выполнение определенных действий, или функций. Эти функции операционная система выполняет, запуская специальные системные программные модули, входящие в ее состав.

В настоящий момент около 90% персональных компьютеров используют ОС Windows, которая имеет ряд достоинств и вытеснила конкурентов из этого сегмента рынка. Более широкий класс ОС ориентирован для использования на серверах. К этому классу ОС относят: семейство Unix, разработки фирмы Microsoft, сетевые продукты Novell и корпорации IBM.

К ресурсами компьютера относятся: процессоры, память, дисковые накопители, сетевые коммуникационные средства, принтеры и другие устройства. Функцией ОС является рациональное распределение этих ресурсов между процессами с целью обеспечения максимальной эффективности функционирования компьютера.

## Практическая часть

## I. Общая характеристика задачи

Исходные данные для расчета заработной платы организации представлены на рис.17.1 и 17.2.

Построить таблицы по приведенным ниже данным.

В таблице на рис.17.3 для заполнения столбцов «Фамилия» и «Отдел» использовать функцию ПРОСМОТР().

Для получения результата в столбец «Сумма по окладу», используя функцию ПРСМОТР(), по табельному номеру найти соответствующий оклад, разделить его на количество рабочих дней и умножить на количество отработанных дней. «Сумма по надбавке» считается аналогично. Данные берутся из графы «Надбавка».

Сформулировать ведомость зарплаты.

Данные результатной таблицы отсортировать по отделам. По результатам вычислений построить круговую диаграмму.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таб. номер | Фамилия | Отдел | Оклад, руб. | Надбавка, руб. |
| 001 | Иванов И.И. | Отдел кадров | 6000,00 | 4000,00 |
| 002 | Петров П.П. | Бухгалтерия | 4500,00 | 3000,00 |
| 003 | Сидоров С.С. | Отдел кадров | 5000,00 | 4500,00 |
| 004 | Мишин М.М. | Столовая | 5500,00 | 3500,00 |
| 005 | Васин В.В. | Бухгалтерия | 6500,00 | 1000,00 |
| 006 | Львов Л.Л. | Отдел кадров | 5000,00 | 3000,00 |
| 007 | Волков В.В. | Отдел кадров | 3000,00 | 3000,00 |

Рис.17.1. Данные о сотрудниках

|  |  |  |
| --- | --- | --- |
| Таб.  номер | Количество  рабочих дней | Количество  отработанных дней |
| 001 | 23 | 23 |
| 002 | 23 | 20 |
| 003 | 27 | 27 |
| 004 | 23 | 23 |
| 005 | 23 | 21 |
| 006 | 27 | 22 |
| 007 | 23 | 11 |

Рис.17.2 Данные об учете рабочего времени

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таб. номер | Фамилия | Отдел | Сумма по окладу, руб. | Сумма по надбавке, руб. | Сумма зарплаты, руб. | НДФЛ,% | Сумма НДФЛ, руб. | Сумма к выдаче, руб. |
|  |  |  |  |  |  | 13 |  |  |
|  |  |  |  |  |  |  |  |  |

Рис.17.3. Графы таблицы для заполнения ведомости зарплаты

Описание алгоритма решения задач.

I. Построить таблицы «Данные о сотрудниках», «Данные об учете свободного времени» и «Ведомости зарплаты» по приведенным данным.

Запустить табличный процессор MS Excel:

нажать кнопку Пуск

выбрать в Главном меню команды Все программы / MS Office / MS Office Excel 2003.

В открывающейся книге (Книга 1) Лист 1 переименовать в лист с наименованием Данные о сотрудниках:

внизу окна Книги 1 щелкнуть два раза левой кнопкой мыши ярлык Лист 1

на выделенном ярлыке набрать Данные о сотрудниках

нажать клавишу Enter

На рабочем листе Данные о сотрудниках MS Excel создать таблицу:

установить курсор мыши на ячейку А1 и щелчком выделить ячейку

набрать заголовки таблицы в одну строку (каждый заголовок вводится в свою самостоятельную ячейку)

выделить все ячейки. Занятые заголовками граф, для чего нажать левую кнопку мыши на первой ячейке и, удерживая кнопку, растянуть выделение по всем необходимым ячейкам.

в строке меню выбрать команду Формат / Ячейки

в диалоговом окне Формат ячеек выбрать Выравнивание

установить выравнивание по горизонтали и вертикали и установить перенос по словам

для сохранения изменений нажать кнопку ОК (рис.1).

(При необходимости можно расширить столбцы: установить курсор мыши на нужной границе столбцов, нажать левую кнопку мыши и, удерживая ее, изменить размер столбца)

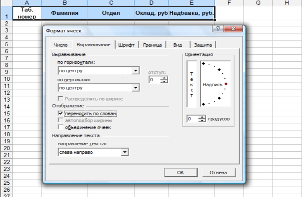


Рис.1 Выбор формата ячеек

Заполнить таблицу - Данные о сотрудниках исходными данными (рис.2):

поместить курсор мыши в нужную ячейку, щелкнуть левой кнопкой мыши один раз.

ввести текст или цифровые данные

подтвердить набор нажатием клавиши Enter

выделив столбец А, выбираем Формат – Ячейки – Число – Дополнительный – Таб. Номер

выделив столбцы D и E, выбираем Формат – Ячейки – Число – Денежный

сделать границы таблицы, нажав значок Границы на панели управления, предварительно выделив всю таблицу



Рис.2 Таблица «Данные о сотрудниках»

В открывающейся книге (Книга 1) Лист 2 переименовать в лист с наименованием Данные об учете рабочего времени:

внизу окна Книги 1 щелкнуть два раза левой кнопкой мыши ярлык Лист 2

на выделенном ярлыке набрать Данные об учете рабочего времени

нажать клавишу Enter

На рабочем листе Данные об учете рабочего времени MS Excel создать таблицу:

установить курсор мыши на ячейку А1 и щелчком выделить ячейку

набрать заголовки таблицы в одну строку (каждый заголовок вводится в свою самостоятельную ячейку)

выделить все ячейки. Занятые заголовками граф, для чего нажать левую кнопку мыши на первой ячейке и, удерживая кнопку, растянуть выделение по всем необходимым ячейкам.

в строке меню выбрать команду Формат / Ячейки

в диалоговом окне Формат ячеек выбрать Выравнивание

установить выравнивание по горизонтали и вертикали и установить перенос по словам

для сохранения изменений нажать кнопку ОК (рис.1).

Заполнить таблицу Данные о сотрудниках исходными данными (рис.3):

поместить курсор мыши в нужную ячейку, щелкнуть левой кнопкой мыши один раз.

ввести текст или цифровые данные

подтвердить набор нажатием клавиши Enter

выделив столбец А, выбираем Формат – Ячейки – Число – Дополнительный – Таб. Номер

сделать границы таблицы, нажав значок Границы на панели управления, предварительно выделив всю таблицу



Рис.3 Таблица «Данные об учете рабочего времени»

В открывающейся книге (Книга 1) Лист 3 переименовать в лист с наименованием Ведомости зарплаты:

внизу окна Книги 1 щелкнуть два раза левой кнопкой мыши ярлык Лист 3

на выделенном ярлыке набрать Ведомости зарплаты

нажать клавишу Enter

На рабочем листе Ведомости зарплаты MS Excel создать таблицу:

установить курсор мыши на ячейку А1 и щелчком выделить ячейку

набрать заголовки таблицы в одну строку (каждый заголовок вводится в свою самостоятельную ячейку)

выделить все ячейки. Занятые заголовками граф, для чего нажать левую кнопку мыши на первой ячейке и, удерживая кнопку, растянуть выделение по всем необходимым ячейкам.

в строке меню выбрать команду Формат / Ячейки

в диалоговом окне Формат ячеек выбрать Выравнивание

установить выравнивание по горизонтали и вертикали и установить перенос по словам

для сохранения изменений нажать кнопку ОК (рис.1).

Заполнить таблицу Данные о сотрудниках исходными данными (рис.4):

поместить курсор мыши в нужную ячейку, щелкнуть левой кнопкой мыши один раз.

ввести текст или цифровые данные

подтвердить набор нажатием клавиши Enter

сделать границы таблицы, нажав значок Границы на панели управления, предварительно выделив всю таблицу

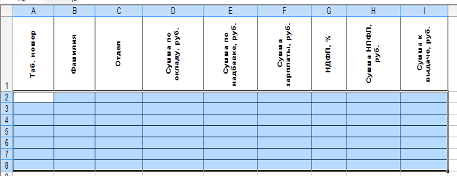


Рис.4 Таблица «Ведомости зарплаты»

## II. Формирование ведомости зарплаты

В столбец А вводим табельные номера и, выделив весь столбец выберем Формат – Ячейки – Число – Дополнительный – Таб. Номер

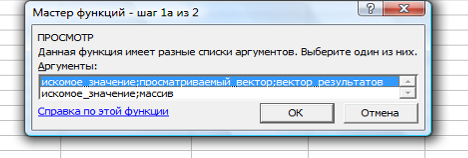
С помощью функции ПРОМОТР() заполнить столбец «Фамилия»

помещаем курсор мыши в ячейку В2

в ячейке выбираем Вставка – Функция – выбираем функцию ПРОСМОТР

для подтверждения нажать клавишу ОК (это был шаг 1)

в появившемся окне Мастер функций – шаг 1а из 2 выбираем отметку искомое значение; просматриваемый вектор; вектор результатов



для подтверждения нажать клавишу ОК в появившемся окно Аргументы функции заполняем соответственно рис.5

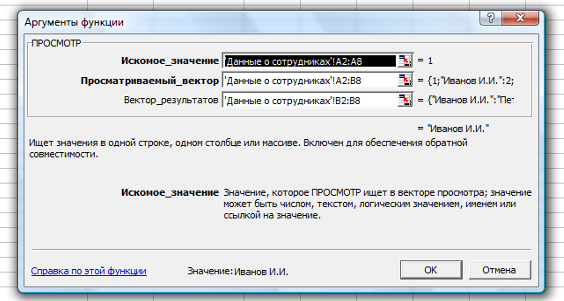


Рис.5

для подтверждения нажимаем ОК

размножить введенную в ячейку В2 формулу для остальных ячеек (с В3 по В8) данной графы

С помощью функции ПРОМОТР() заполнить столбец «Отдел»

столбец «Отдел» заполнить аналогично столбцу «Фамилия»

диалоговое окно Аргументы функции заполняем соответственно Рис.6

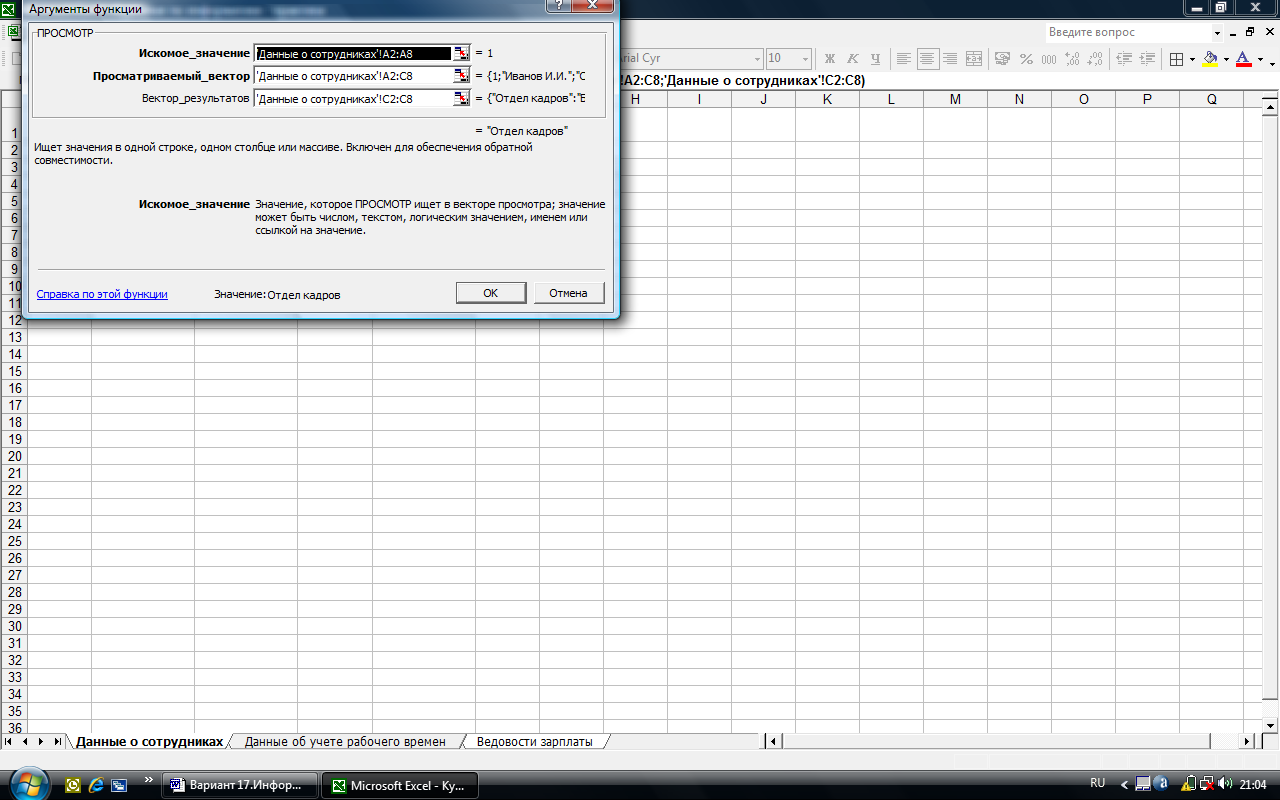


Рис.6

Для получения результата в столбце «Сумма по окладу», используя функцию ПРОСМОТР(), по табельному номеру найти соответствующий оклад, разделить его на количество рабочих дней и умножить на количество отработанных дней. «Сумма по надбавке» считается аналогично. Данные берутся из графы «Надбавка» (для столбцов В, С, D, E, F, H, I выбрать Формат – Ячейки – Число – Денежный)

помещаем курсор мыши в ячейку D2 и вводим =ПРОСМОТР('Данные о сотрудниках'! A2: A8; 'Данные о сотрудниках'! A2: D8; 'Данные о сотрудниках'! D2: D8) /ПРОСМОТР('Данные о сотрудниках'! A2: A8; 'Данные об учете рабочего времен'! A11: B17; 'Данные об учете рабочего времен'! B11: B17) \*ПРОСМОТР('Данные о сотрудниках'! A2: A8; 'Данные об учете рабочего времен'! A11: C17; 'Данные об учете рабочего времен'! C11: C17)

размножить введенную в ячейку D2 формулу для остальных ячеек (с D3 по D8) данной графы

в ячейку Е2 вводим =ПРОСМОТР('Данные о сотрудниках'! A2: A8; 'Данные о сотрудниках'! A2: E8; 'Данные о сотрудниках'! E2: E8) /ПРОСМОТР('Данные о сотрудниках'! A2: A8; 'Данные об учете рабочего времен'! A11: B17; 'Данные об учете рабочего времен'! B11: B17) \*ПРОСМОТР('Данные о сотрудниках'! A2: A8; 'Данные об учете рабочего времен'! A11: C17; 'Данные об учете рабочего времен'! C11: C17)

размножить введенную в ячейку E2 формулу для остальных ячеек (с E3 по E8) данной графы

Заполнить столбец «Сумма зарплаты»

помещаем курсор мыши в ячейку F2

в каждой ячейки вводим =D2+E2

размножить введенную в ячейку F2 формулу для остальных ячеек (с F3 по F8) данной графы

Для столбца G выбираем формат Процентный и вводим число 13.

Заполнить столбец «Сумма НДФЛ»

помещаем курсор мыши в ячейку H2

в каждой ячейки вводим =F2\*G2

размножить введенную в ячейку H2 формулу для остальных ячеек (с H3 по H8) данной графы

8. Заполнить столбец «Сумма к выдаче»

помещаем курсор мыши в ячейку I2

в каждой ячейки вводим =F2-H2

размножить введенную в ячейку H2 формулу для остальных ячеек (с I3 по I8) данной графы

Таблица примет вид, показанный на рис.7

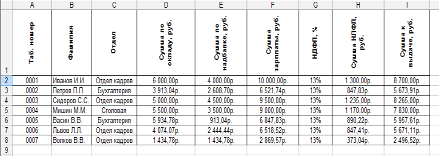


Рис.7 Таблица «Ведомости зарплаты».

## III. Данные результатной таблицы

Отсортировать по номеру отдела и рассчитать итоговые суммы по отделам. По результатам вычислений построить круговую диаграмму.

Данные результатной таблицы отсортировать по номеру отдела

вставить новый Лист, для этого щелкнуть правой кнопкой между листами и выбрать добавить – Лист

выделить таблицу «Ведомость зарплаты»

копировать таблицу на новый Лист 1 (только значения)

выделить таблицу на Листе 1

выбрать в Главном меню Данные / Сортировка

в появившемся окне в поле Сортировать по указать Отдел и Идентифицировать поля по подписям

нажать кнопку ОК (рис.8)

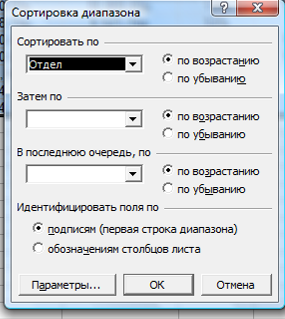


Рис.8 Сортировка диапазона

Таблица примет вид, показанный на рис.9

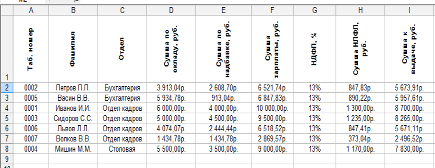


Рис.9

Рассчитать итоговые суммы по отделам

выбрать в Главном меню Данные / Сводная таблица

в окне Мастер сводных таблиц и диаграмм шаг 1 выбрать Сводная таблица

нажать кнопку Далее

в появившемся окне Мастер сводных таблиц и диаграмм шаг 2 выбрать диапазон А1: I8 и нажать кнопку Далее

в окне Мастер сводных таблиц и диаграмм шаг 3 выбрать отметку Поместить в новом листе, нажать кнопку Макет.

в окне Мастер сводных таблиц и диаграмм – Макет перетащить поля соответственно рис.10

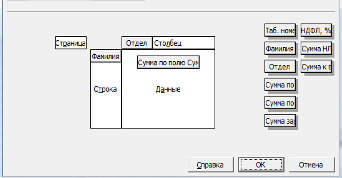


Рис.10. Сводная таблица

Построить круговую диаграмму

на Листе 1 выделить ячейки В2: В8, затем, удерживая Ctrl, выделить ячейки I2: I8

выбрать в Главном меню Вставка – Диаграмма и выбрать Круговую и нажать ОК.

в появившемся окне Мастер диаграмм (шаг 2 из 4): источник данных диаграмм выбрать диапазон, указанный на рис.11, и ряды в столбцах.

для подтверждения нажать Далее

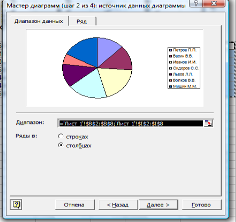


Рис.11. Мастер диаграмм – шаг 2

в появившемся окне Мастер диаграмм (шаг 3 из 4): источник данных диаграмм (рис.12) нажать Далее

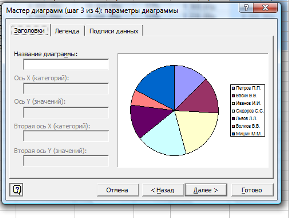


Рис.12. Мастер диаграмм – шаг 3

в появившемся окне Мастер диаграмм (шаг 4 из 4): источник данных диаграмм (рис.13) Поместить диаграмму на листе и выбрать Отдельном

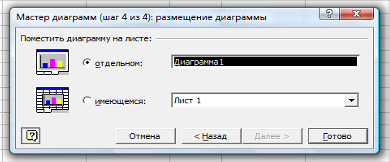


Рис.13. Мастер диаграмм – шаг 4

В итоге диаграмма имеет вид, показанный на рис.15

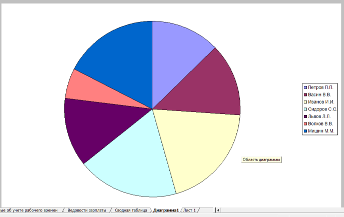


Рис.14. Круговая диаграмма

## Список использованной литературы

1. Гордеев А.В. Операционные системы: Учебник для вузов.2-е изд. – СПб.: Питер, 2005.
2. Основы информатики: Учеб. пособие / А.Н. Морозевич, Н.Н. Говядинова, В.Г. Левашенко и др.; Под ред.А.Н. Морозевича. – 2-е изд., испр. – Мн.: Новое знание, 2003.
3. Евсюков В.В. Экономическая информатика: Учеб. пособие – Тула: Издательство «Граф и К», 2003.
4. Информатика в экономике: Учеб. пособие /Под ред. проф. Б.Е. Одинцова, проф.А.Н. Романова. – М.: Вузовский учебник, 2008.

1. Виртуальная машина – функциональный эквивалент воображаемого компьютера с заданной конфигурацией, моделируемый программно-аппаратными средствами реального компьютера. [↑](#footnote-ref-1)
2. Прерывание – прекращение вычислительного процесса, вызванное требованиями на обслуживание иных устройств компьютера. [↑](#footnote-ref-2)
3. Если операционная система не поддерживает механизм учетных записей, как это имеет место, например, в семействе операционных систем Windows 9x компаний Microsoft, то пароль сверяется по специальному файлу, где он хранится в зашифрованном виде. [↑](#footnote-ref-3)
4. В 70-е годы пользователи за терминалом писали log in, и это означало процедуру регистрации. Были системы, в которых требовалось набрать команду log on, что означало то же самое. [↑](#footnote-ref-4)