**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение** **высшего профессионального образования**

**"Национальный исследовательский университет"**

**"Высшая школа экономики"**

Московский институт электроники и математики

Национального исследовательского университета

«Высшая школа экономики»

Департамент прикладной математики

**Домашнее задание**

**по дисциплине**

“Операционные системы и компьютерные сети”

**Часть 1**

**Работу выполнила**

**студент 3 курса**

**группы БПМ-192**

Пчелинцев Илья

**Преподаватель**

Внуков Андрей Анатольевич

Москва, 2021

**Оглавление**

[Введение](#bookmark=id.gjdgxs)…………………………………………………………………………3

[Постановка задачи](#bookmark=id.30j0zll)………………………………………………………………3

[Основная часть](#bookmark=id.3znysh7)…………………………………………………………………..3

[Литература](#bookmark=id.2p2csry)………………………………………………………………………5

**Введение**

**Операционная система** – программный комплекс, который позволяет использовать аппаратуру компьютера и предоставляет интерфейс для управления и реализации прикладного программного обеспечения.

Операционная система является обязательным компонентом любой вычислительной машины, какие бы задачи перед ней ни стояли — будь то домашний компьютер, узел локальной или глобальной компьютерной сети, сервер баз данных или же комплекс управления технологическим процессом на промышленном предприятии.

Операционная система должна быть достаточно прозрачной для разработчиков программного обеспечения, дабы те могли разрабатывать приложения для расширения функционала операционной системы и улучшения ее работы. Поэтому любая операционная система базируется на совокупности нескольких важных понятий и концепций.

**Постановка задачи**

В данном задании требуется раскрыть следующие темы:

1. [Расслоение памяти](#bookmark=id.tyjcwt)
2. [Мультипроцессорная обработка](#bookmark=id.2jxsxqh)
3. [Компиляторы](#bookmark=id.4i7ojhp)
4. [Интерпретаторы](#bookmark=id.2xcytpi)

**Основная часть**

1. **Расслоение памяти**. Метод расслоения памяти применяется для увеличения скорости доступа к оперативной памяти компьютера. В общем случае во время обращения к одной из ячеек оперативной памяти никакие другие обращения к памяти производиться не могут. При расслоении памяти соседние по адресам ячейки размещаются в различных блоках так, что появляется возможность производить несколько обращений одновременно. Быстродействие увеличивается.

Известны два основных метода расслоения памяти. Суть этих методов состоит в том, что память строится на основе нескольких модулей. Но в одном случае модули памяти имеют раздельные адресные пространства (независимая адресация), а в другом – модули охвачены общим полем адресов и образуют единое адресное пространство.

Метод 1. Метод разделения памяти на два модуля с независимой адресацией был предложен и опробован еще в конце 50-х гг. лабораториями Гарвардского университета. Один модуль памяти использовался для хранения команд программы, другой – данных. Оба модуля имели собственные контроллеры памяти и раздельные магистрали доступа. Такой принцип построения ОП оказался во многих случаях очень эффективным и был успешно использован при разработке компьютеров различного назначения. Возник даже термин "Гарвардская архитектура". Этот принцип сохранен и в современных вариантах ЭВМ (процессорах) Гарвардской архитектуры. Он позволяет совместить во времени циклы обмена с обоими модулями памяти и оказывается эффективным при обработке любых типов программ. В настоящее время Гарвардская архитектура широко используется при построении кэш-памятей мощных процессоров.

Метод 2. В простейшем случае используют два модуля с "веерной" (чередующейся) адресацией, при которой смежные адреса информационных единиц, соответствующих ширине выборки (слово, двойное слово и т.д.), принадлежат разным модулям (т.е. четные адреса принадлежат одному модулю, а нечетные – другому). Это позволяет процессору инициировать второй цикл обмена до завершения первого, поскольку адреса лежат в разных модулях, либо обращаться одновременно двум устройствам к разным модулям памяти. В результате за счет перекрытия во времени обращений к разным модулям пропускная способность ОП в среднем повышается. Веерная адресация, как и конвейер команд, оказывается эффективной только при наличии в программе достаточно длинных участков с последовательным выбором команд.

Двунаправленное расслоение ОП не может предотвратить появления циклов ожидания, поэтому были разработаны системы с 4- и более кратным расслоением ОП, в которых контроллер памяти обеспечивает распределение последовательно вырабатываемых адресов между несколькими модулями памяти. [■](#bookmark=id.1fob9te)

1. **Мультипроцессорная обработка.** Мультипроцессорная обработка - это способ организации вычислительного процесса в системах с несколькими процессорами, при котором несколько задач (процессов, потоков) могут одновременно выполняться на разных процессорах системы.

При мультипроцессорной работе возникает опасность конфликтных ситуаций определенных типов, которых не бывает в однопроцессорных машинах. Здесь необходимо обеспечить координированный упорядоченный доступ к каждой общей ячейке памяти, с тем чтобы два процессора не могли изменять ее содержимое одновременно — и в результате, быть может, портить его. Подобная координация необходима также и в случае, когда один процессор пытается изменить содержимое ячейки, которую хочет прочитать другой процессор. Упорядочение доступа необходимо также и для однопроцессорных машин. [■](#bookmark=id.1fob9te)

1. **Компиляторы.** Программирование на языке ассемблера требует больших усилий и, чтобы облегчить процесс программирования, были разработаны языки высокого уровня. Программы теперь можно писать, не тратя время и силы на полное изучение внутренней структуры компьютера. Языки высокого уровня позволили безболезненно переносить программы с одного компьютера на другой (другого типа). Перевод программ с языка высокого уровня на машинный язык осуществляется при помощи программы, называемой компилятором. Перевод осуществляется по следующей схеме: исходная программа – объектная программа – компоновка – загрузочная программа. [■](#bookmark=id.1fob9te)
2. **Интерпретаторы.** Интерпретаторы – особый вид трансляторов, которые не генерируют объект-программу, а фактически обеспечивают непрерывное выполнение исходной программы. Выполнение программы в режиме интерпретации происходит медленнее по сравнению с компилированным кодом, поскольку интерпретатору надо транслировать команды при каждом ее выполнении. Пример: команда1, команда2, команда3, команда1, команда 3 (последние две транслируются заново). [■](#bookmark=id.1fob9te)

**Литература**

1. Wikipedia.org

2. Таненбаум С. Современные ОС.- С.-П.: Изд. Группа BHV, 2002

3. Стивене Р.У., Раго С.А. UNIX. Профессиональное программирование. 2-е издание. – СПб.: Символ-Плюс, 2007 г.