Министерство науки и высшего образования Российской Федерации

Государственное образовательное учреждение высшего профессионального образования

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Н.Э. БАУМАНА

(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления» Кафедра «Компьютерные системы и сети»

Утверждаю

Зав.каф. ИУ6

\_\_\_\_\_\_\_\_\_\_Пролетарский А.В.

«\_\_\_»\_\_\_\_\_\_\_ 2022 г.

Большие вычислительные машины (мейнфрейм):

архитектура и микроархитектура zSeries,

операционные системы

и

основы программирования

Методические указания

по выполнению лабораторных работ

по курсу БЭВМ

Издание четвертое, исправленное

2022 год

Москва

1

Содержание

[**Работа в среде БЭВМ с использованием эмуляторов мейнфрейма Hercules и терминала Vista TN3270, установленных на сервере Ubuntu v20.04 с ОС LINUX**](#_heading=h.p8qf7go4mvrq) **4**

[1 Подключение ко внутренней сети МГТУ (vpn)](#_heading=h.td0ic6iqkejo) 4

[2 Подключение к серверу кафедры и к эмулятору мейнфрейма Hercules ( z/OS)](#_heading=h.ro4gxi7mx715) 7

[2.1 Установка эмулятора терминала TN 3270 и подключение его к мейнфрейму](#_heading=h.iplz5495nvq) 8

[2.2 Вход в операционную систему z/OS](#_heading=h.yksdcorhxamt) 9

[3 Создание наборов данных (Data Set) в zOS](#_heading=h.lcwpgqq0zian) 13

[4 Выход из операционной системы z/OS (z/OS Logoff)](#_heading=h.79h0f4l9j0pj) 24

[5 Контрольные вопросы](#_heading=h.htol237y0grw) 26

[**Редактирование программ в среде MAINFRAME c использованием редактора ISPF и эмулятора мейнфрейма Hercules**](#_heading=h.lk6yk0ejxamc) **28**

[1 Вход в операционную систему z/OS](#_heading=h.qdcqhk5yyoqk) 28

[2 Подготовка к разработке программы на языке С](#_heading=h.xqaacm77xsxf) 29

[3 Редактирование данных в режиме ISPF Editor (кодирование на языке С)](#_heading=h.8qojaczfnbj2) 29

[4 Создание JCL-скрипта и его выполнение](#_heading=h.104kp85i9rhk) 32

[5 Поиск ошибок и работа с утилитой SDSF](#_heading=h.ri6ray5im1jw) 36

[6 Выход из операционной системы z/OS (z/OS Logoff)](#_heading=h.slr76b12eqpi) 38

[7 Контрольные вопросы](#_heading=h.pt5xabn5vf7j) 40

2

Министерство науки и высшего образования Российской Федерации

Государственное образовательное учреждение высшего профессионального образования

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Н.Э. БАУМАНА

(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра «Компьютерные системы и сети»

Смирнова Е.В., Сапарбаев Т.Р., Бояринов А.С.,

Дубровин Е.Н., Шевченко В.А.

Лабораторная работа №1

**Работа в среде БЭВМ с использованием эмуляторов мейнфрейма Hercules и терминала TN3270**

Методические указания по выполнению лабораторной работы

по курсу по выбору «Архитектура больших ЭВМ»

2022 год,

Москва

## Работа в среде БЭВМ с использованием эмуляторов мейнфрейма Hercules и терминала Vista TN3270, установленных на сервере Ubuntu v20.04 с ОС LINUX

**Цели лабораторной работы № 1:**

1. Освоение способов подключения к мейнфрейму с использованием эмулятора большой ЭВМ Hercules и работа в операционной системе z/OS.
2. Создание контейнеров данных (Data Set) для хранения трех файлов (Member) разных типов.

**Выполнение лабораторной работы**

### 1 Подключение ко внутренней сети МГТУ (vpn)

В первую очередь необходимо настроить и подключить Open VPN.

1. Скопируйте в адресную строку браузера путь к инсталлятору (см. рисунок 1)

<https://openvpn.net/community-downloads/>.

****

Рисунок 1 – Open VPN

1. Выберите для Вашей разрядности Windows, загрузите и запустите его на своей машине, как показано на рисунке 2.

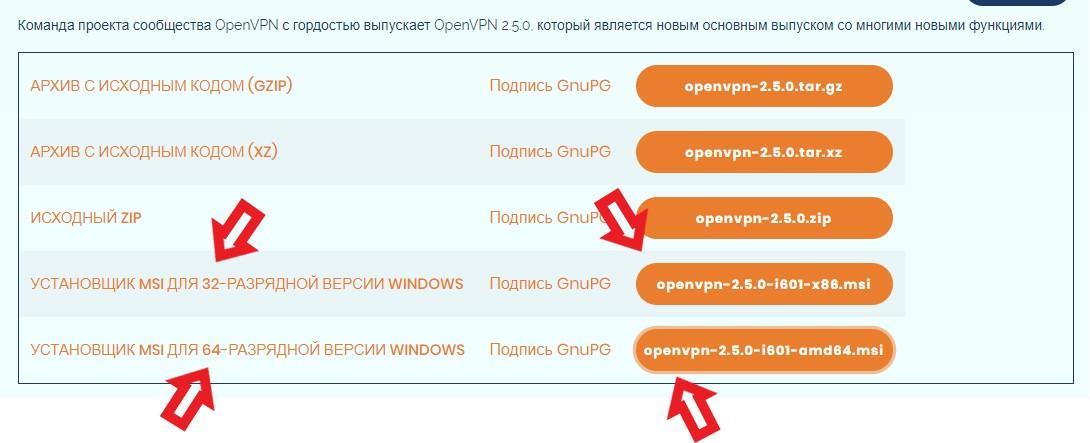


Рисунок 2 – выбор инсталлятора

1. Подтвердите установку vpn, как показано на рисунке 3.

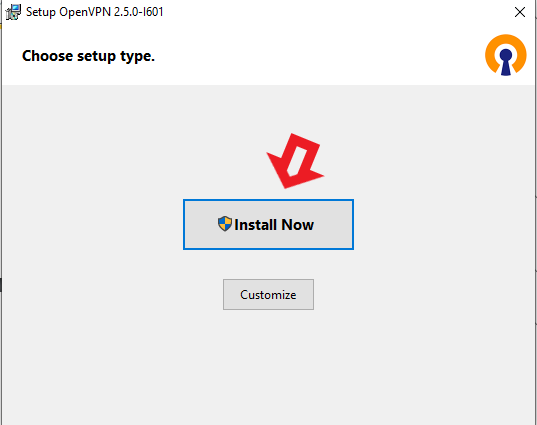
****

Рисунок 3 – Установка

1. Скачайте архив с конфигурацией

<https://mail.bmstu.ru/~postmaster/bmstu.ovpn> , как показано на рисунке 4.

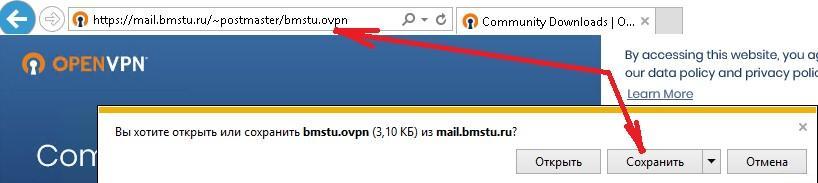
****

Рисунок 4 – Скачивание конфигурации для OpenVPN

1. Найдите в системном трее иконку программы OpenVPN, щелкните по ней правой кнопкой мыши и выберите пункт «Импорт конфигурации …» , как показано на рисунке 5.

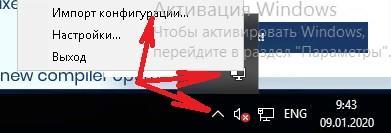


Рисунок 5 – Импорт конфигурации

1. Найдите в разделе «Загрузки» (или любой другой папке, куда вы сохраняете загруженные файлы) файл bmstu.ovpn и откройте его, как показано на рисунке 6.

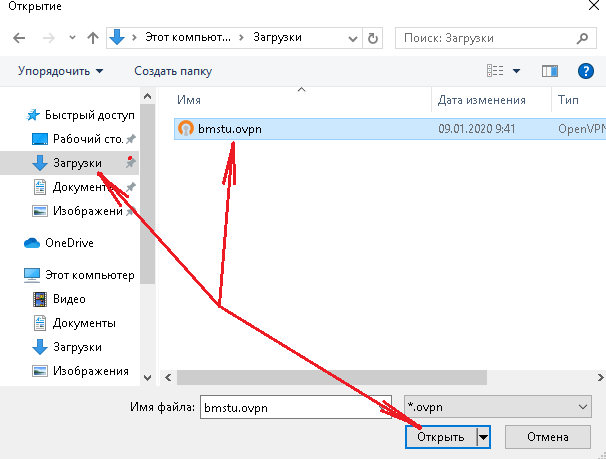
****

Рисунок 6 – Конфигурация для подключения ко внутренней сети МГТУ им. Н.Э. Баумана bmstu.ovpn

1. Подключение. Щелкните правой кнопкой мыши по изображению программы в трее и выберите пункт «Подключиться», как показано на рисунке 7.

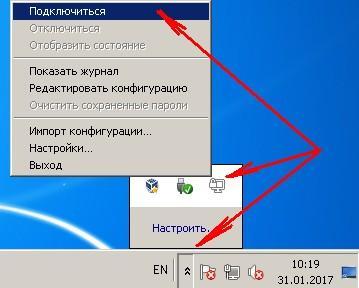


Рисунок 7 – Подключение через системный трей

1. Авторизация. Укажите свой логин без @bmstu.ru, пароль к электронной почте университета и нажмите на кнопку ОК, как показано на рисунке 8.

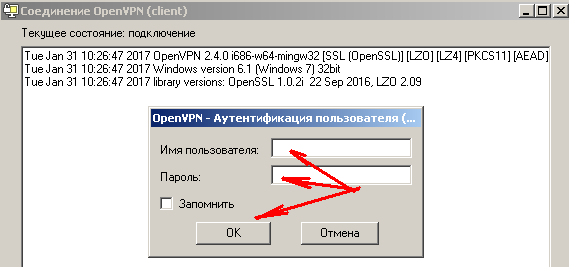
****

Рисунок 8 – авторизация

Теперь Вы подключены ко внутренней сети МГТУ и можете работать с сервером кафедры ИУ-6.

### 2 Подключение к серверу кафедры и к эмулятору мейнфрейма Hercules ( z/OS)

Лабораторная работа выполняется на эмуляторе мейнфрейма Hercules — это программный эмулятор мейнфреймов IBM ([System/370](https://ru.wikipedia.org/wiki/System/370),  [System/390](https://ru.wikipedia.org/wiki/System/390),  [zSeries / System z](https://ru.wikipedia.org/wiki/System_z)) и совместимых ([Amdahl](https://ru.wikipedia.org/w/index.php?title=Amdahl&action=edit&redlink=1)). Эмулятор мейнфрейма Hercules запускается на персональных компьютерах с ОС [Linux](https://ru.wikipedia.org/wiki/Linux),  [Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows),  [FreeBSD](https://ru.wikipedia.org/wiki/FreeBSD),  [Solaris](https://ru.wikipedia.org/wiki/Solaris), [Mac OS X](https://ru.wikipedia.org/wiki/Mac_OS_X), [MorphOS](https://ru.wikipedia.org/wiki/MorphOS). Его код распространяется по условиям O[pen Source](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5). Эмулирует лишь центральный процессор и небольшое количество периферийных устройств. Hercules стал первым эмулятором 64-битных мейнфреймов [z/Architecture](https://ru.wikipedia.org/wiki/Z/Architecture).

Операционная система z/OS поддерживает множество способов обращения к ней. Для того чтобы получить доступ к серверу z/OS с вашего компьютера, вам необходимо установить компонент доступа, так называемый Клиент. Примеры таких компонентов показаны на рисунке 9.

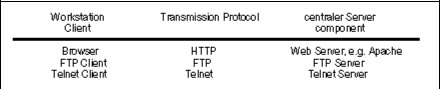


Рисунок 9 – компоненты доступа к z/OS

В частности, вы можете использовать протокол IBM 3270, который обеспечивает связь с сервером коммуникаций z/OS. Этот протокол использует протокол TELNET, как носитель, и осуществляет доступ через TCP/IP порт 23. Клиент 3270 называют также эмулятором терминала Mainframe (см. рисунок 10).

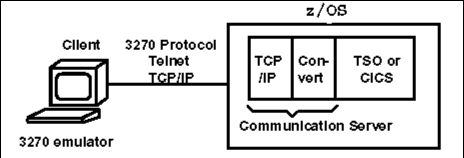


Рисунок 10 – Клиент 3270 и сервер z/OS

#### 2.1 Установка эмулятора терминала TN 3270 и подключение его к мейнфрейму

Эмулятор терминала мейнфрейма TN3270 вы можете скачать отсюда: <https://www.mochasoft.dk/tn3270.htm> (предоставляется 30 дней бесплатного использования).

Окно запуска эмулятора терминала TN3270 дано на рисунке 11.

Перед входом в TN3270 не забудьте подключить vpn (см. раздел 1 Подключение ко внутренней сети МГТУ).

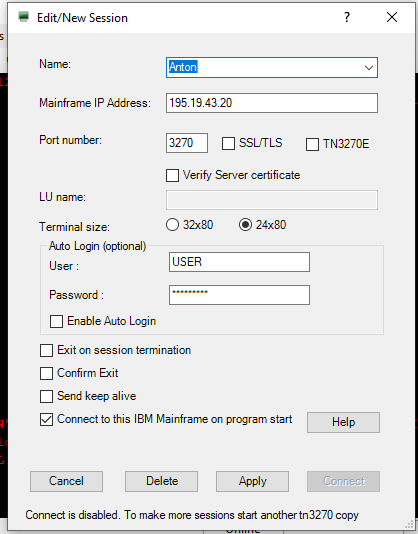


Рисунок 11 – Окно интерфейса эмулятора терминала – настройка сессии.

Установите и запустите эмулятор терминала, указав адрес сервера

**IP - 195.19.43.20 и Port 3270 (ВНИМАНИЕ! Номер порта может быть другой).**

**Name – свое имя на Английском.**

**Логин и пароль получите у преподавателя.**

#### 2.2 Вход в операционную систему z/OS

При удачном соединении на экране появится окно приветствия операционной системы z/OS (см. рисунок 12) и машина будет ожидать от Вас ввода Вашего логина (для работы в операционной системе z/OS Вы должны получить регистрационный номер (UserID) и пароль (Password) у преподавателя).

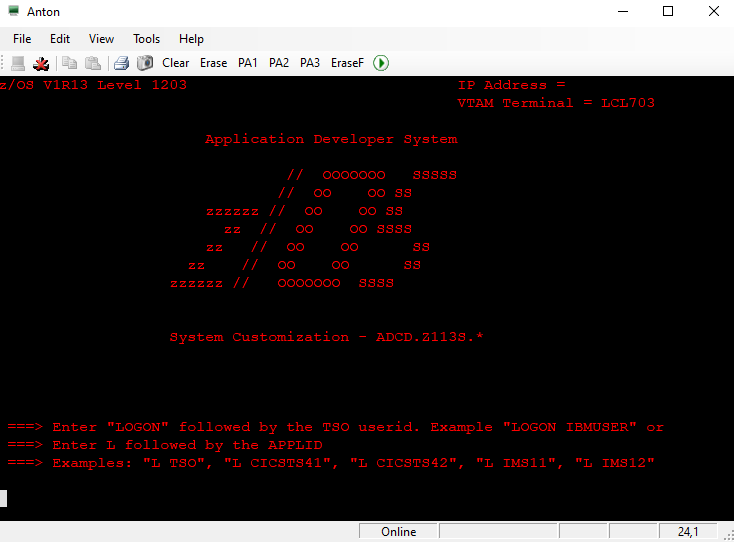


Рисунок 12 – Окно приветствия системы z/OS

На этом месте ВНИМАНИЕ!

ЧИТАЙТЕ ВНИМАТЕЛЬНО ВСЕ СООБЩЕНИЯ ОТ МЕЙНФРЕЙМА!

Введите в input команду LOGON и следуйте инструкции по connection user (пока не добавлены другие пользователи). После ввода Вашего логина появится окно (рисунок 13.1).

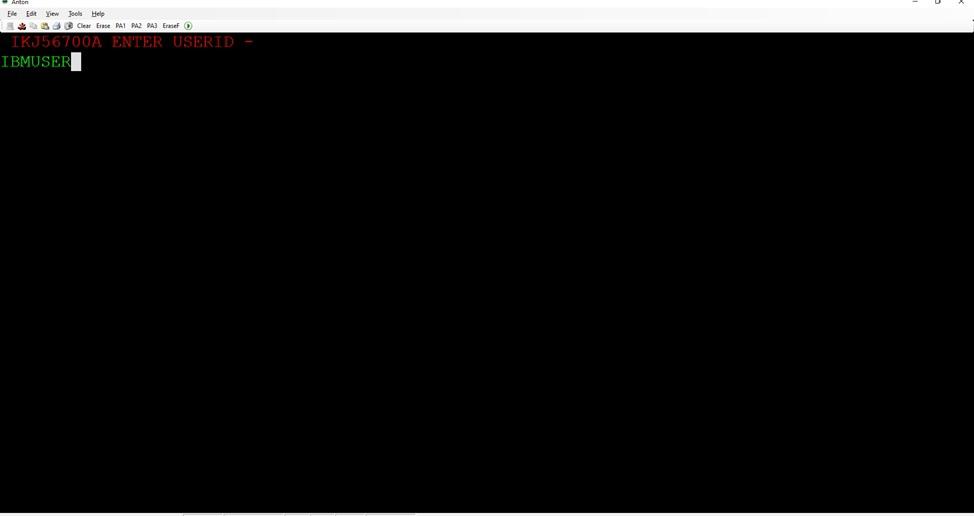


Рисунок 13.1 – Окно входа в операционную систему TSO/E. Ввод UserID

После ввода UserID перед вами появится приветственный экран **TSO/E** (рисунок 13.2). Введите пароль Password и нажмите еще раз **Enter,** и вы попадете в приветственный экран системы TSO ( рисунок 14).

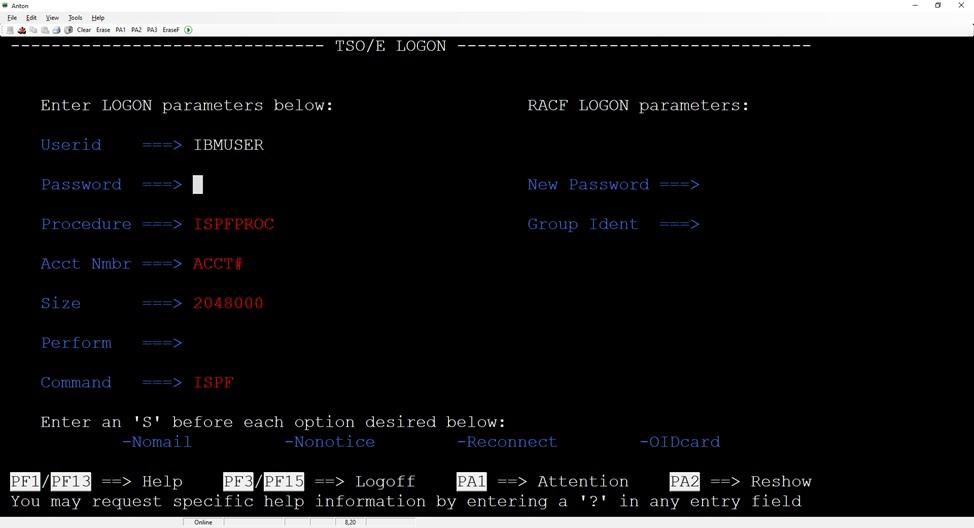


Рисунок 13.2 – Окно входа в операционную систему TSO/E. Ввод пароля

При первом запуске система попросит Вас сменить пароль на новый. Введите новый пароль поле **New Password** (не больше 7 символов!!!) и нажмите **Enter**. Подтвердите новый пароль, напечатав его в поле **New Password** еще раз, и нажмите **Enter**. Не забудьте его! Программа “Восстановление пароля” на этой машине не запущена (на апрель 2022).

**Символ \*\*\* (три звезды)** говорит о том, что система готова вывести следующее сообщение. Каждый раз, когда Вы видите на экране символ “\*\*\*”, Вы можете не торопиться и прочитать сообщения системы, по нажатию клавиши Enter система выдаст продолжение информационного текста.

**Примечание: Это - важная информация!** Если вы будете сдавать тест на получение Сертификата системного администратора мейнфрейма - знайте, это ответ на один из вопросов теста.

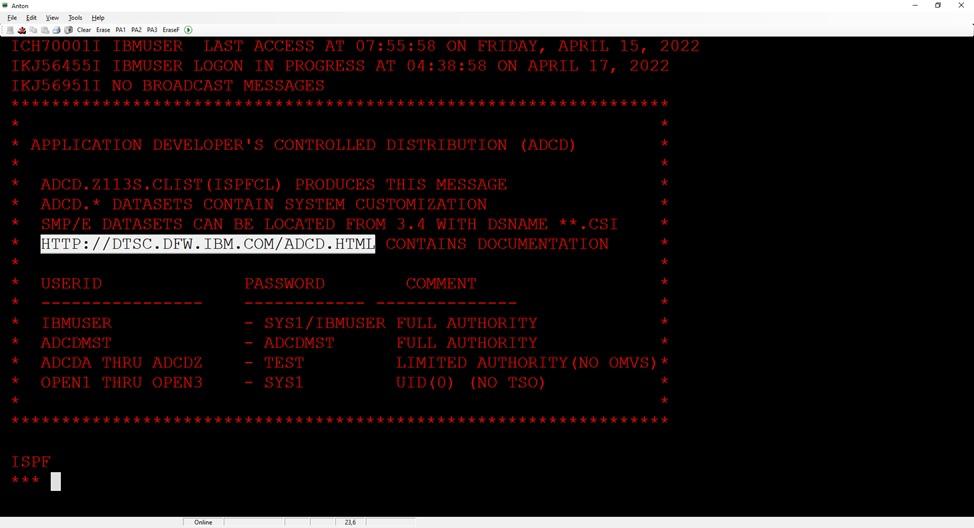


Рисунок 14 – Окно входа в ISPF

Нажмите еще раз **Enter,** и вы попадете в окно интерфейса **ISPF** (рисунок 15). Если не попали, нажмите F3 и введите слово (команду) ISPF.

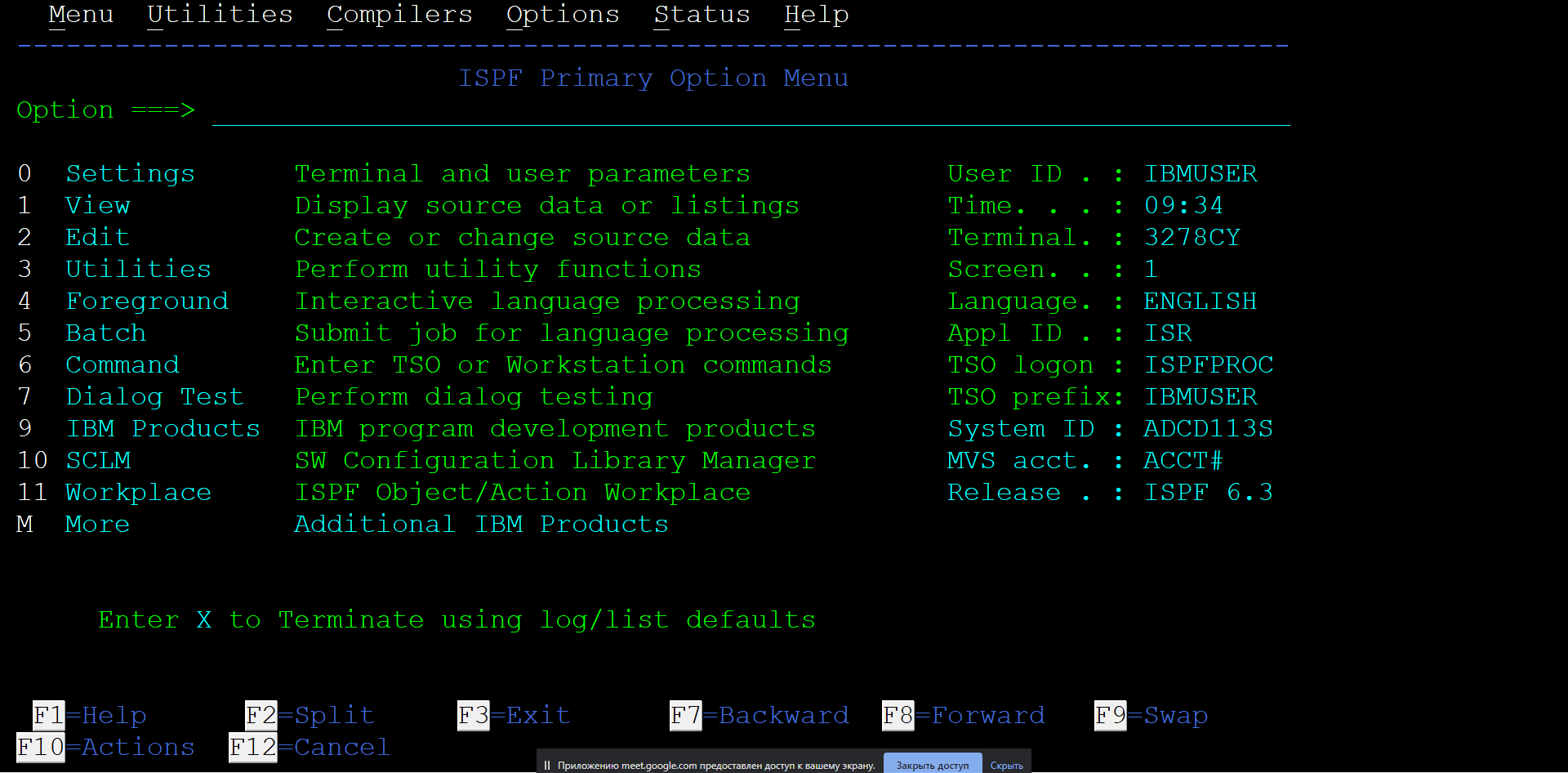


Рисунок 15 – Начальное окно интерфейса ISPF

Подсистема ISPF построена как иерархическое меню, позволяющее вызывать различные функции TSO, вводя команды в командной строке. Операционная система z/OS управляет многими подсистемами. Иерархическая взаимосвязь подсистем и функций, которые вы будете осваивать, показана на рисунке 16.

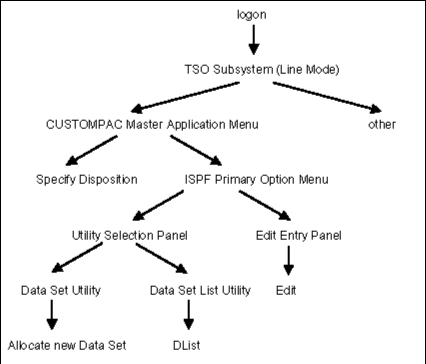


Рисунок 16 – Иерархическая структура подсистем и функций ISPF

Для того, чтобы перейти к выбранному пункту меню, введите номер пункта в командной строке. Если вам нужно перейти сразу на несколько уровней меню вниз, вы можете ввести номера пунктов, разделенные точкой. Так, введя «3.2» в главном меню ISPF (рисунок 15), вы попадете в утилиту Data Set из меню Utilites. Чтобы вернуться назад из подменю или утилиты, нажмите **F3**.

**Примечание:** **Это - важная информация!** Если вы будете сдавать тест на получение Сертификате сисадмина мейнфрейма - знайте, это ответ на один из вопросов теста.

### 3 Создание наборов данных (Data Set) в zOS

Теперь создадим собственный набор данных (Data Set).

**Набор данных** (data set) - именованная совокупность связанных элементов данных, размещаемых во внешней памяти или иных устройствах. Операционная система z/OS поддерживает работу с наборами данных различной логической организации: последовательными, прямого доступа, библиотечными (PDS и PDSE) и некоторыми другими. Мы будем использовать библиотечный тип набора данных PDS (Partitioned Data Set), так как он наиболее похож на «папку» OC Windows.

Библиотечные наборы данных PDS - это совокупность разделов (members), снабженная оглавлением (directory), в котором содержится информация об именах разделов и их размещении в памяти.

Структура набора данных распределенного типа (Partitioned Data Set) показана на рисунке 17. Такие наборы данных обычно используются для хранения относительно небольших по объему "блоков" информации: исходных текстов программ, процедур и заданий, объектных модулей, текстовых документов, таблиц и т.п.

| Directory | Member 1 | Member 2 | Member 3 | Member 4 | ... |
| --- | --- | --- | --- | --- | --- |



Рисунок 17 – Структура набора данных распределенного типа (Partitioned Data Set)

Для создания набора данных из основного окна интерфейса **ISPF** (рисунок 15) войдем в меню **Utility Selection Panel**, нажав «3» в командной строке (рисунок 18).

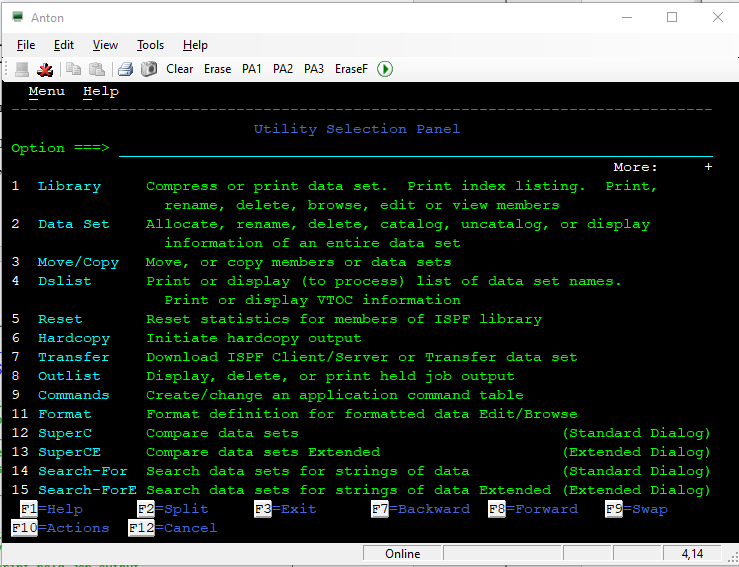


Рисунок 18 – Окно панели Utility Selection Panel

Для того чтобы создавать, редактировать и запускать на выполнение программы, нам надо создать три набора данных (Data Set) для хранения:

1. текстовых файлов – исходных кодов, текстов программ и др.;
2. JCL-скриптов (аналог UNIX makefile), которые инструктируют z/OS о компилировании и связях исходных кодов;
3. исполняемых файлов (машинных кодов).

Сейчас мы создадим первый набор данных для тренировки, остальные наборы данных Вы создадите самостоятельно в конце лабораторной работы.

Введите «2» – появится окно «**Data Set Utility**», как показано на рисунке 19.

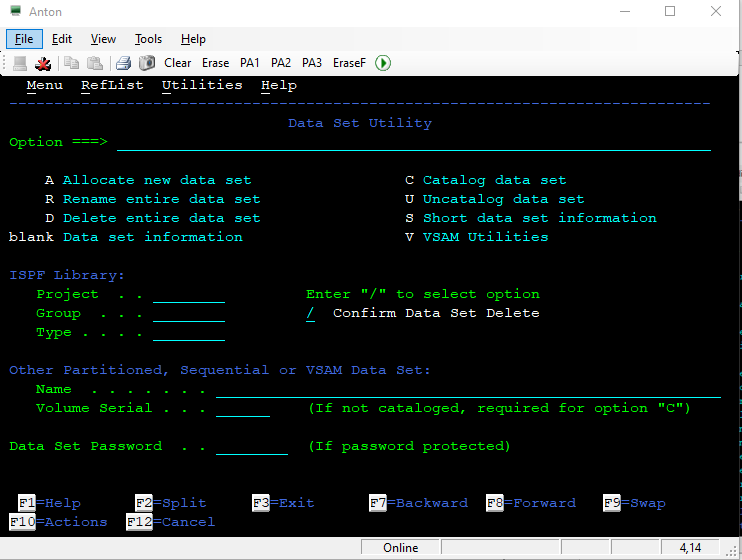


Рисунок 19 – Окно Data Set Utility

Интерфейс ISPF предполагает, что мы введем имя набора данных, который хотим создать. Имена бывают простые и составные. Простое имя может содержать не более 8 символов (латинские буквы A-Z, цифры 0-9, спецсимволы #,@,$,-), причем первым символом имени не может быть цифра. Например, РАRTS01, B1934-1, $$$$A. Составное имя набора данных складывается из нескольких простых, разделенных символом "." ("точка"). Например, D.USER1.JCL, А.VERY.LONG.DATASET.NАМЕ, $PARTS.DАTА2. Имя файла-раздела формируется по тем же правилам и указывается в круглых скобках после имени библиотечного набора данных, например: MY.DSET.PROG(PROG01) - раздел имя файла (Member) PROG01 набора данных MY.DSET.PROG.

Для начала создадим наш набор данных с именем в формате xxx.yyy.zzz.

В строку option вводим «2»

Далее:

PROJECT xxx (Ваш UserID)

GROUP TEST

TYPE C

В этом материале для создания файлов используется пользовательский UserID IBMUSER. При выполнении лабораторной работы вместо него вы должны будете использовать ваш собственный UserID, выданный вам преподавателем.

В командной строке «Option===>» введите «A» и нажмите Enter. В результате откроется следующее окно подсистемы распределения пространства New Data Set Allocation (рисунок 20).

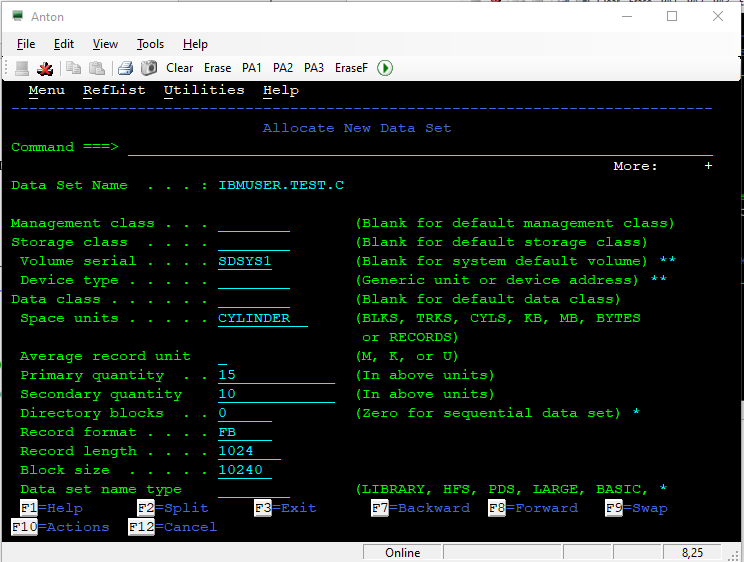


Рисунок 20– Окно определения нового набора данных

Мы хотим создать набор данных, в который будут помещены данные, используя редактор LE (Language Environment). Мы выбрали имя набора данных «**xxx.TEST.С**» и ввели его в три поля, как показано на рисунке 20.

**Примечание**: Бывают случаи, когда TSO и ISPF требуют, чтобы параметры были введены заглавными буквами. Поэтому желательно всегда использовать прописные буквы при работе с TSO, ISPF и CICS.

*Размер набора данных DS*

Определим размеры нашего набора данных. Будем использовать MEGABYTE в качестве единицы измерения размера набора данных. Также могут использоваться альтернативные единицы размеров – tracks, cylinders, и другие. Мы определили максимальный размер в 2 MEGABYTE (см. рисунок 21).

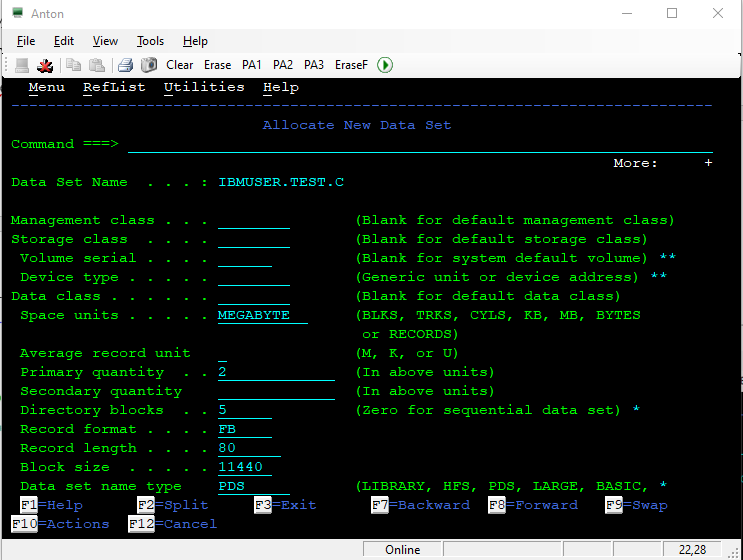


Рисунок 21 – Окно определения нового набора данных с введенными параметрами

Тип набора данных – библиотечный (PDS). Для этого поставим цифру «**5**» в строке «**Directory Blocks**» (разным типам наборов данных соответствуют разные цифры - например, «**0**» будет обозначать тип «**Sequential Data Set**», «**5**» - «**Partitioned Data Set**»). Также непосредственно укажем тип набора данных в поле «**Data Set Name Type**» - «**PDS**». Введенные вами данные должны совпадать с теми, что показаны на рисунке 21, кроме строки имени.

*Структура наборов данных (Data Set)*

Немного о структуре наборов данных. В z/OS набор данных представляется в виде совокупности логических записей, а приложения получают доступ к логическим записям и обрабатывают их, как единое целое. В то же время обмен данными между периферийными устройствами и основной памятью (ввод-вывод) осуществляются блоками (или физическими записями). В блоке объединяется некоторое количество логических записей, как показано на рисунке 22. Таким образом, для каждого набора данных необходимо установить согласованные размеры логических записей (LRECL) и блоков (BLKSIZE). То есть, размер блока должен быть получен умножением длины записи на определенное число, например: 80 х 143 = 11440.

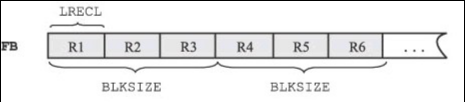


Рисунок 22 – Набор данных с записями фиксированной длины

*Формат записи*

В z/OS поддерживаются три формата логических записей: записи фиксированной длины, записи переменной длины, записи неопределенной длины. Записи фиксированной длины имеют постоянный размер и идентифицируются символами **F** или **FB** в зависимости от выбранного способа блокирования записей: **F** - в каждом блоке содержится только одна логическая запись, **FB** - каждом блоке может содержаться более одной логической записи.

При выполнении лабораторной работы формат записи (**Record**) определим как «**FB**» (**Fixed Blocks** – фиксированные блоки), длина записи (**Record Length**) – «**80 Bytes**» и размер блока – **11440** записей. Выбор длины записи в 80 байт не случаен – он оптимален для представления текстовых файлов, так как это длина строки в терминале и на перфокарте :) Остальные поля заполняются автоматически.

Введите параметры нового набора данных и нажмите **Enter**, как показано на рисунке 23.

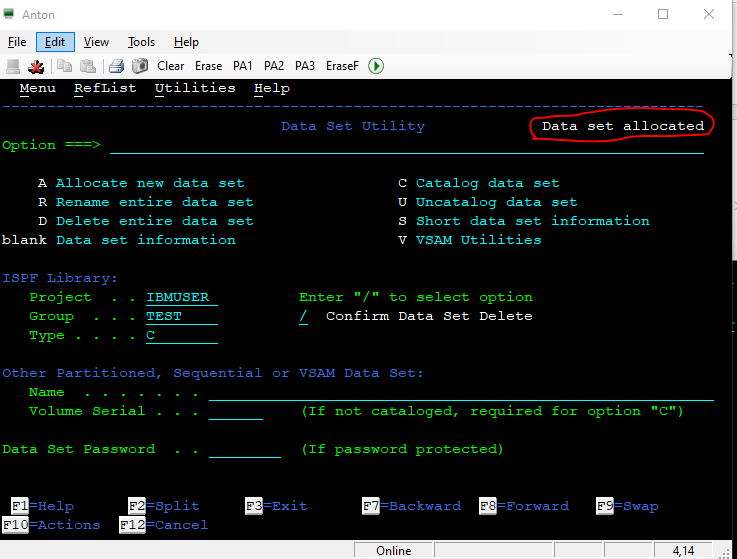


Рисунок 23 – Новый набор данных создан (allocated)

В правом верхнем углу (рисунок 23) вы увидите сообщение о том, что новый набор данных был определен (allocated). Наш набор данных способен теперь включить в себя несколько файлов – разделов (members). Нажмите F3 и вернитесь в панель «Utility Selection Panel» (см. рисунок 24).

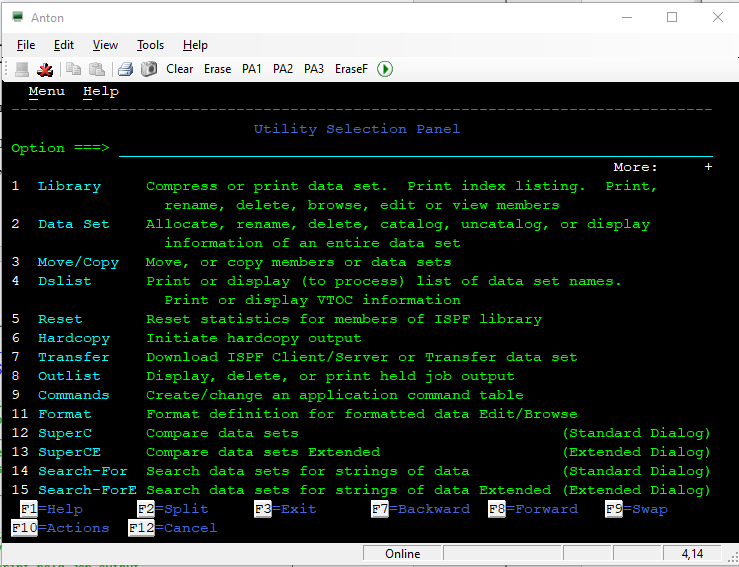


Рисунок 24 – Панель Utility Selection Panel

Давайте проверим результат нашей работы. Это можно сделать с помощью функции **Dslist** (**Data Set List**), введите цифру «**4**» в командной строке и нажмите **Enter** (см. рисунок 24).

Окно «Data Set List Utility» предоставляет много возможностей по просмотру и редактированию наборов данных (см. рисунок 25). Если в поле Dsname Level не установлен ваш UserID, то нужно ввести его вручную. Если в поле Dsname Level уже есть ваш UserID, просто нажмите Enter.

Выведется полный список пользовательских наборов данных – Вы увидите созданный Вами набор данных (см. рисунок 26).

*Индивидуальное задание*

Повторите шаги и создайте еще два набора данных: xxx.TEST.CNTL и xxx.TEST.LOAD, где xxx – это ваш UserID. Для создания xxx.TEST.CNTL – просто повторите предыдущие действия, набор данных не будет отличаться по параметрам (см. рисунок 27).

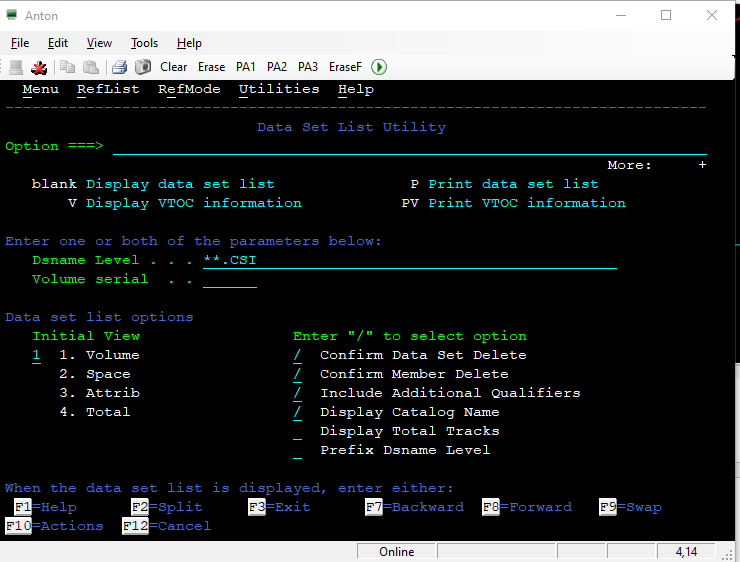


Рисунок 25 – Окно Data Set List Utility

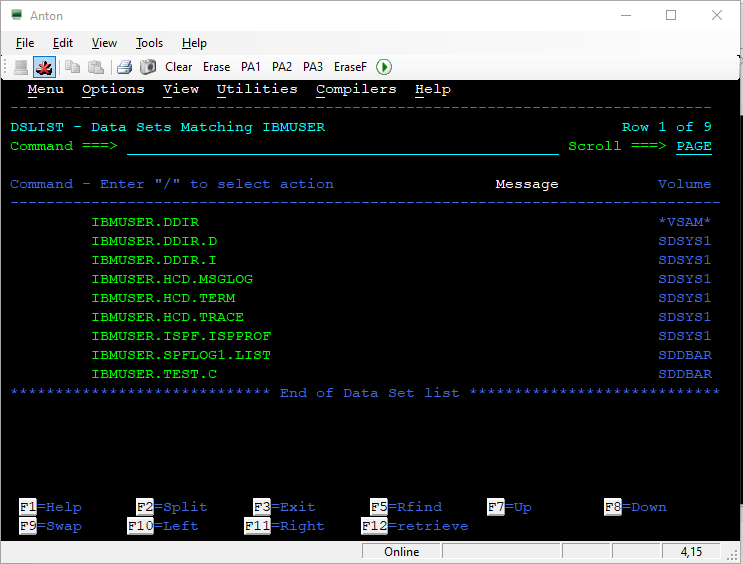


Рисунок 26 – Результат работы – созданный набор данных для файлов

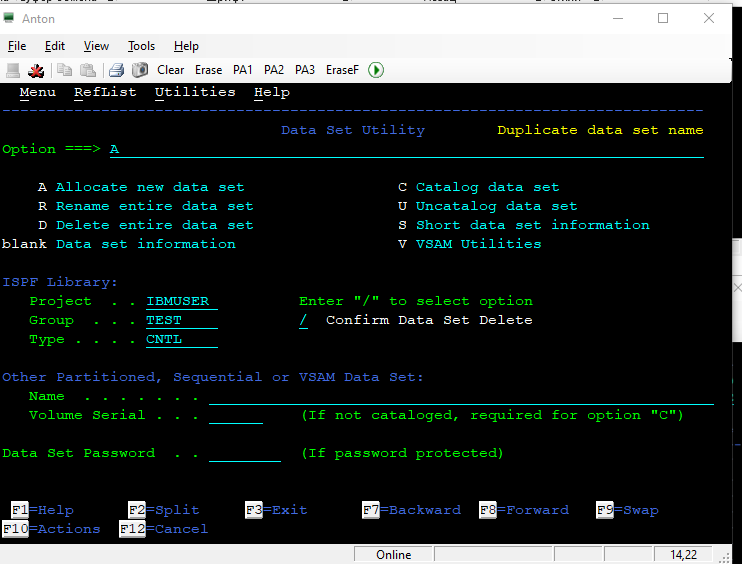


Рисунок 27 – Создание набора данных xxx.TEST.CNTL.

Обратите внимание (см. рисунок 28) на формат записи для набора данных, в котором будут храниться выполняемые программы в машинных кодах, xxx.TEST.LOAD - он должен быть задан как “U” (c записями неопределенной длины). Отметим также, что в этом случае длину записи указывать не обязательно - при использовании записей неопределенной длины (формат U) система не поддерживает деления набора данных на логические записи и производит его обработку блоками фиксированного размера.

После создания еще двух контейнеров xxx.TEST.LOAD и xxx.TEST.CNTL результат вашей работы будет выглядеть так, как показано на рисунке 29.

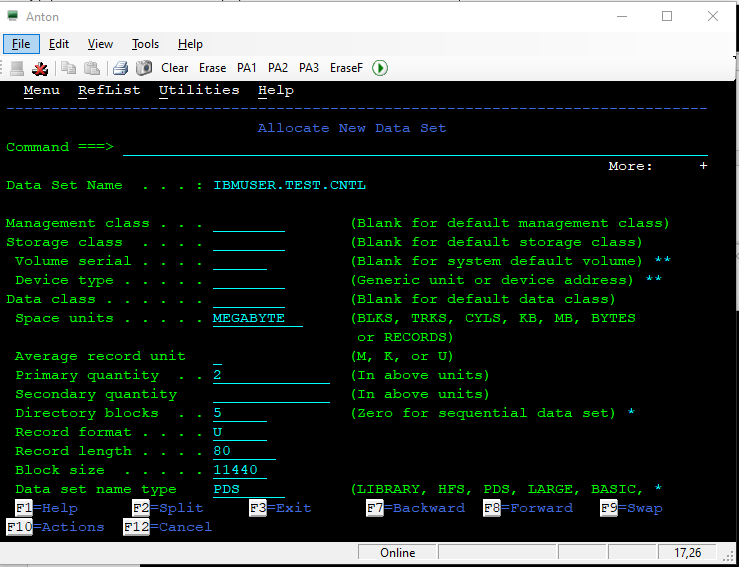


Рисунок 28 – Окно определения нового контейнера (набора данных) xxx.TEST.LOAD с введенными параметрами

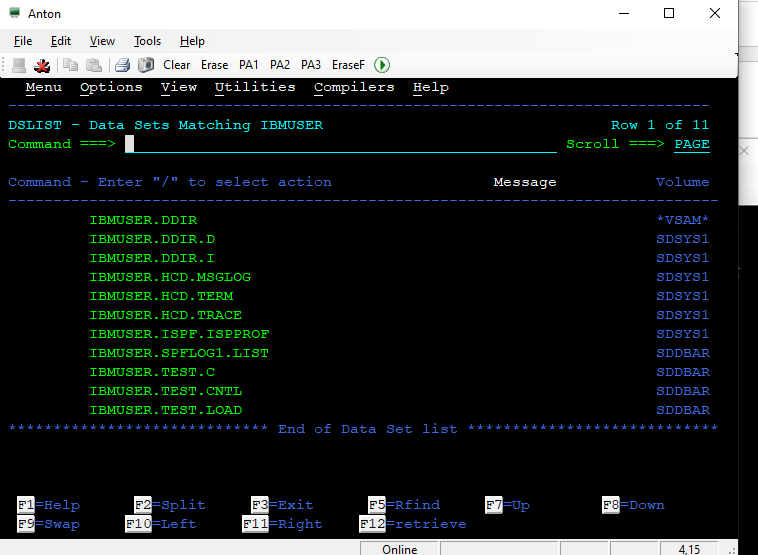


Рисунок 29 – Результат запроса DSLIST

### 4 Выход из операционной системы z/OS (z/OS Logoff)

Нажмите несколько раз клавишу **F3 – шаги назад**, чтобы вернуться в основное окно ISPF. Очередное нажатие клавиши **F3** приведет вас в окно «**Specify Disposition of Data Set**». Система хочет знать, что вы собираетесь сделать с теми наборами данных, которые хранятся в журнале событий, и предоставляет вам на выбор список действий (см. рисунок 30). Выберите «**3**».

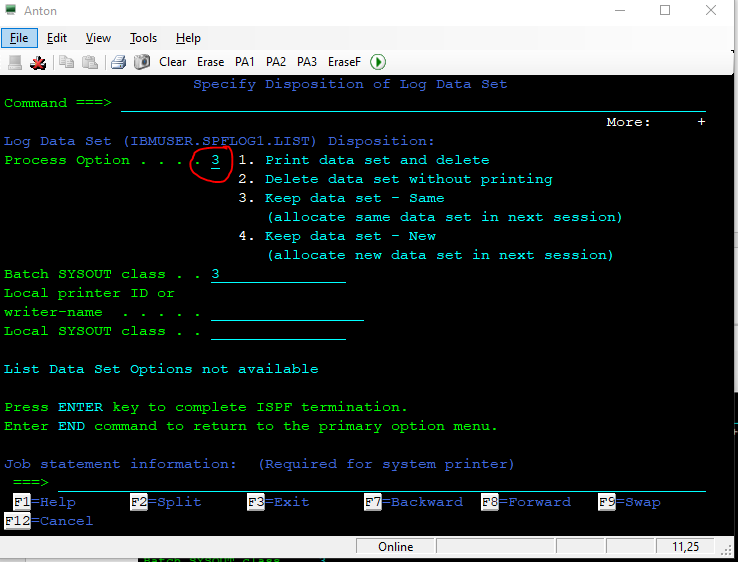


Рисунок 30 – Окно Specify Disposition of Data Set

Произойдет сохранение данных. Появится сообщение TSO о том, что созданный набор данных будет доступен для работы при следующем обращении к системе (см. рисунок 31). При этом набор данных xxx.SPFLOG1.LIST, на который ссылается система, был создан ею автоматически.

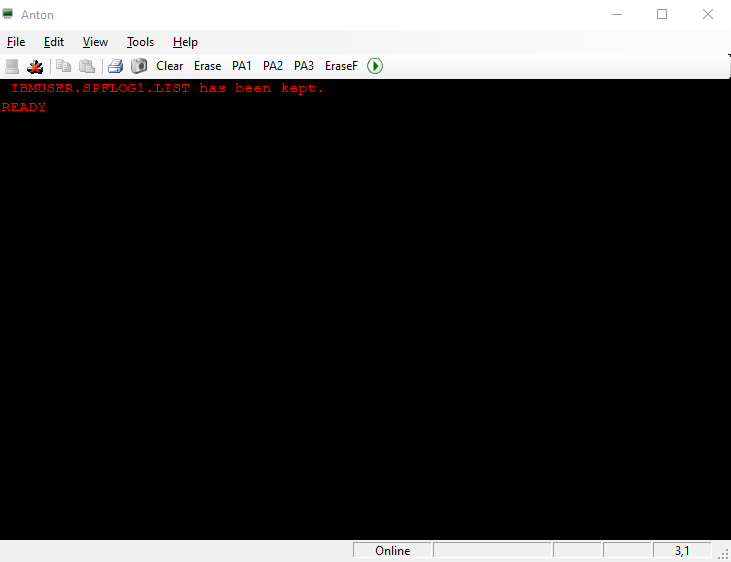


Рисунок 31 – Сообщение TSO

Сообщение «READY» означает ожидание команды от пользователя. Введите слово «LOGOFF» и нажмите Enter.

**Важно**: Если этого не сделать, а просто закрыть окно эмулятора, сессия останется незавершенной, и вы не сможете зайти в систему вновь, т.к. формально вы все еще находитесь в ней. В этом случае придется подождать 30 минут. По истечению таймаута вы снова сможете зайти в систему.

Эмулятор 3270 выведет вас опять в окно приветствия операционной системы z/OS, ожидающее от вас команд.

На этом вы можете закончить сессию связи с виртуальной средой эмулятора HERCULES большой вычислительной машины MAINFRAME.

ПОЗДРАВЛЯЕМ ВАС! ВЫ СДЕЛАЛИ ПЕРВЫЙ ШАГ В НОВУЮ СРЕДУ! Первая лабораторная работа выполнена!

### 5 Контрольные вопросы

5.1 Какие способы обращения к операционной среде z/OS вы знаете?

5.2 Что такое эмулятор 3270?

5.3 Что такое TSO?

5.4 Что такое ISPF?

5.5 Перечислите основные функции ISPF.

5.6 Какую структуру имеют подсистемы и функции ISPF?

5.7 Перечислите последовательность шагов по созданию набора данных.

5.8 Какие типы наборов данных вы знаете?

5.9 Какие поля характеристик набора данных заполняются операционной системой автоматически, и в каких случаях,

5.10 Какая функция ISPF отвечает за просмотр наборов данных?

.11 Как выйти из режима ISPF?

Министерство науки и высшего образования Российской Федерации

Государственное образовательное учреждение высшего профессионального образования

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. Н.Э. БАУМАНА

(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления» Кафедра «Компьютерные системы и сети»

Утверждаю

Зав.каф. ИУ6

\_\_\_\_\_\_\_\_\_\_Пролетарский А.В.

«\_\_\_»\_\_\_\_\_\_\_ 2022 г.

Галямова Е.В., Дубровин Е.Н., Смирнова Е.В., Сапарбаев Т.Р.,

Бояринов А.С., Дубровин Е.Н., Шевченко В.А.

Лабораторная работа № 2

**Редактирование программ в среде Mainframe с**

**использованием редактора ISPF и эмулятора мейнфрейма Hercules**

Методические рекомендации по курсу «Архитектура больших ЭВМ»

2022 год,

Москва

## Редактирование программ в среде MAINFRAME c использованием редактора ISPF и эмулятора мейнфрейма Hercules

**Цели лабораторной работы:**

1. освоение навыков редактирования разделов (файлов) набора данных с использованием интерфейса редактирования ISPF;
2. создание программы на С в редакторе ISPF;
3. компиляция ее с помощью скрипта JCL;
4. выполнение созданной программы, вызвав программу из редактора ISPF или используя команды TSO.

**Выполнение лабораторной работы**

В этом материале для создания файлов используется пользовательский UserID IBMUSER. При выполнении лабораторной работы №2 вместо него вы должны будете использовать ваш собственный UserID, выданный вам преподавателем.

### 1 Вход в операционную систему z/OS

Войдите в TSO и вызовите окно интерфейса ISPF (см. рисунок 1).

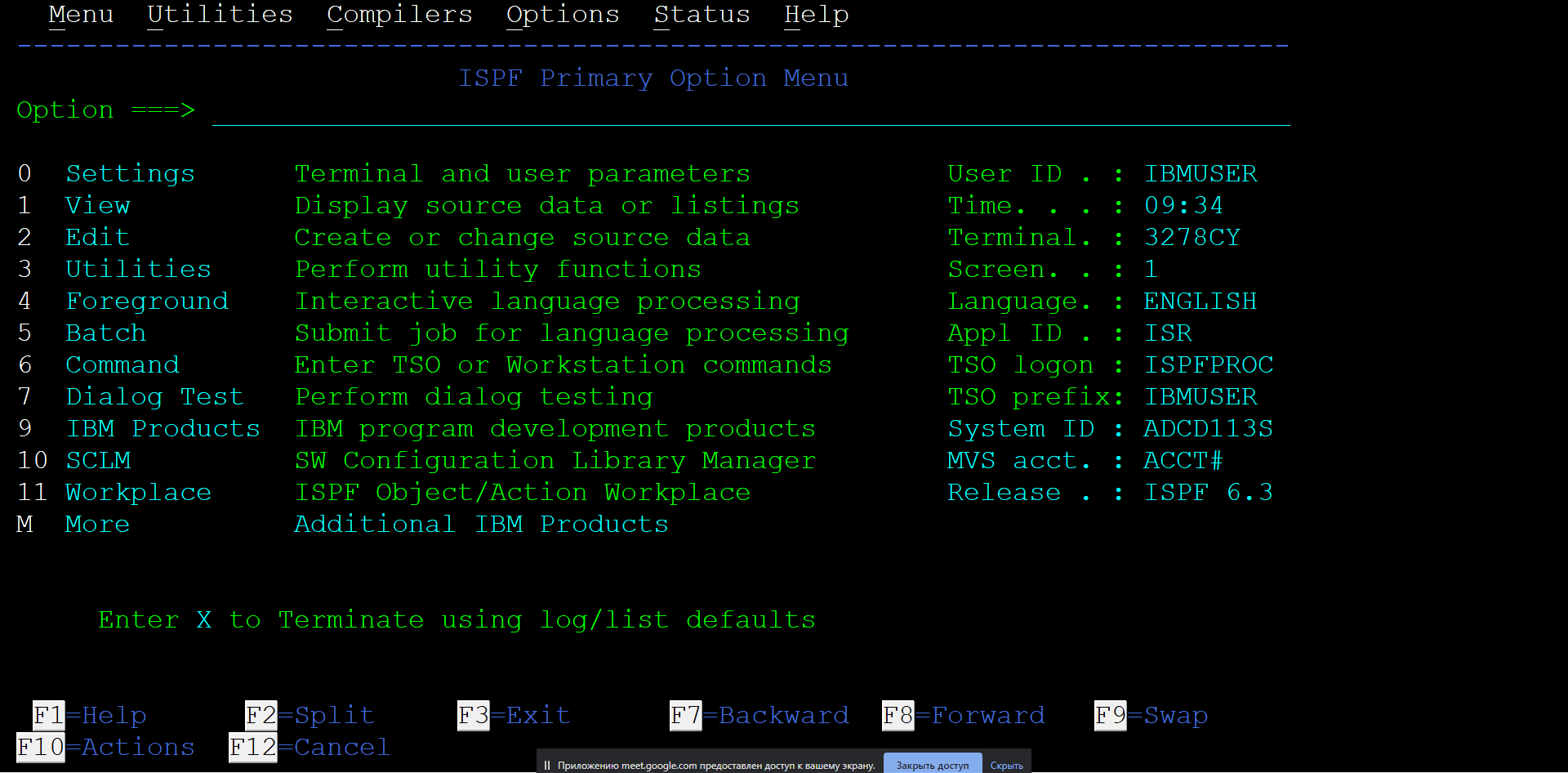


Рисунок 1 – Начальное окно интерфейса ISPF

### 2 Подготовка к разработке программы на языке С

После выполнения лабораторной работы №1 у вас должно быть создано три набора данных:

xxx.TEST.С – для хранения программ в исходных кодах;

xxx.TEST.СNTL – для хранения компилированных программ;

xxx.TEST.LOAD – для хранения программ в машинных кодах (выполняемых),

где xxx – ваш UserID.

Если какой-либо из наборов данных отсутствует, необходимо создать его способом, описанным в лабораторной работе №1.

### 3 Редактирование данных в режиме ISPF Editor (кодирование на языке С)

Вызвать режим редактирования LE (Language Environment) вы можете из основного экрана ISPF, введя «2».

Среда редактирования LE (Language Environment) предоставляет общие службы и языковые подпрограммы в единой среде выполнения. Она обеспечивает согласованные и предсказуемые результаты для ваших приложений, независимо от языка, на котором они написаны. Языковая среда поддерживает совместимость объекта / загрузочного модуля компилятора VS FORTRAN, что означает, что загрузочные модули FORTRAN могут запускаться в языковой среде, а объектный код может быть связан с языковой средой и выполняться в ней. Среда Language Environment также предоставляет набор макросов ассемблера для запуска процедур языка ассемблера. Среда Language Environment поддерживает интерактивный инструмент отладки для отладки приложений в родной среде z/OS, но не является обязательным. Интерактивный IBM Debug Tool доступен с последним выпуском компилятора PL/I, или этот продукт можно заказать отдельно для использования с компиляторами IBM XL C/C++, COBOL и PL/I в z/OS.

Давайте создадим программу на языке С, используя редактор ISPF. В окне подпрограммы редактирования ISPF вы должны ввести имя файла-раздела (Member), который будет хранить исходный код программы. Исходный код программы будет храниться в библиотечном наборе данных xxx.TEST.С, который вы создали в первой лабораторной работе. Раздел, в котором хранится код, должен иметь имя - назовем его V1 (Version 1). Полное имя набора данных - **xxx.TEST.С(V1)**. Введите эти значения в соответствующих полях, как показано на рисунке 2 и нажмите Enter.

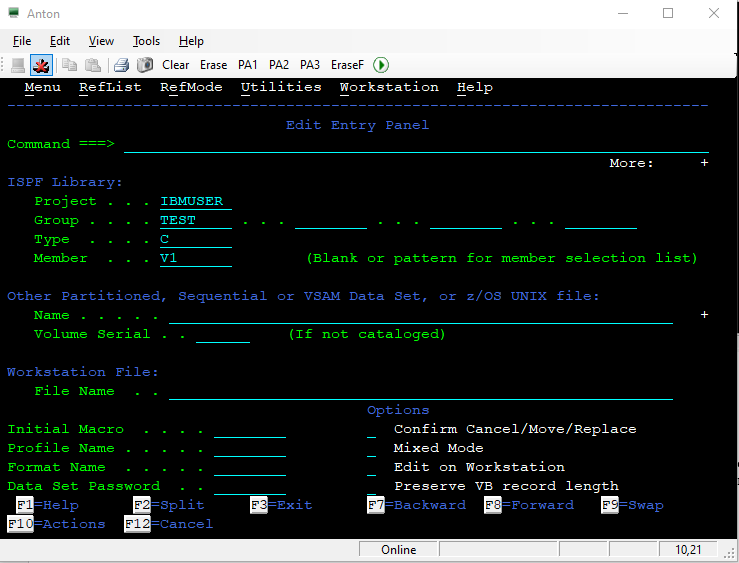


Рисунок 2 – Окно входа в режим редактирования Edit Entry Panel.

После того, как будет выделена память для нового набора данных, экран редактирования будет иметь вид, показанный на рисунке 3.

Используйте цифро-буквенные клавиши, чтобы ввести текст, и не пользуйтесь никакими управляющими клавишами, кроме клавиши Del. Курсор можно двигать клавишами со стрелками.

Чтобы вставить строку, поставьте курсор в левое поле (область номеров строк), в месте, где вы хотите вставить пустую строку, и введите команду Ixx (Insert), где xx – количество строк, которые нужно вставить. Если понадобится удалить строку, введите в левом поле D (Delete) и нажмите Enter. Нажав клавишу F3, вы выйдете из редактора с сохранением файла. Наберите программу, как показано на рисунке 4.

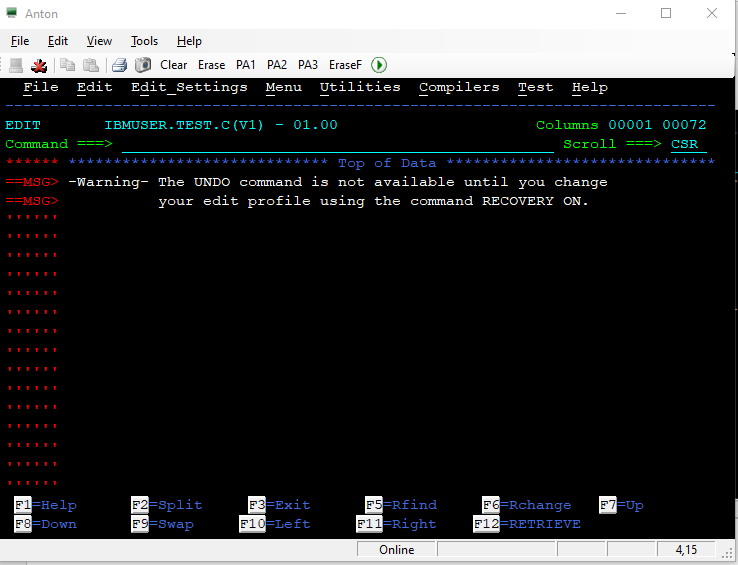


Рисунок 3 – Пустое окно редактирования ISPF

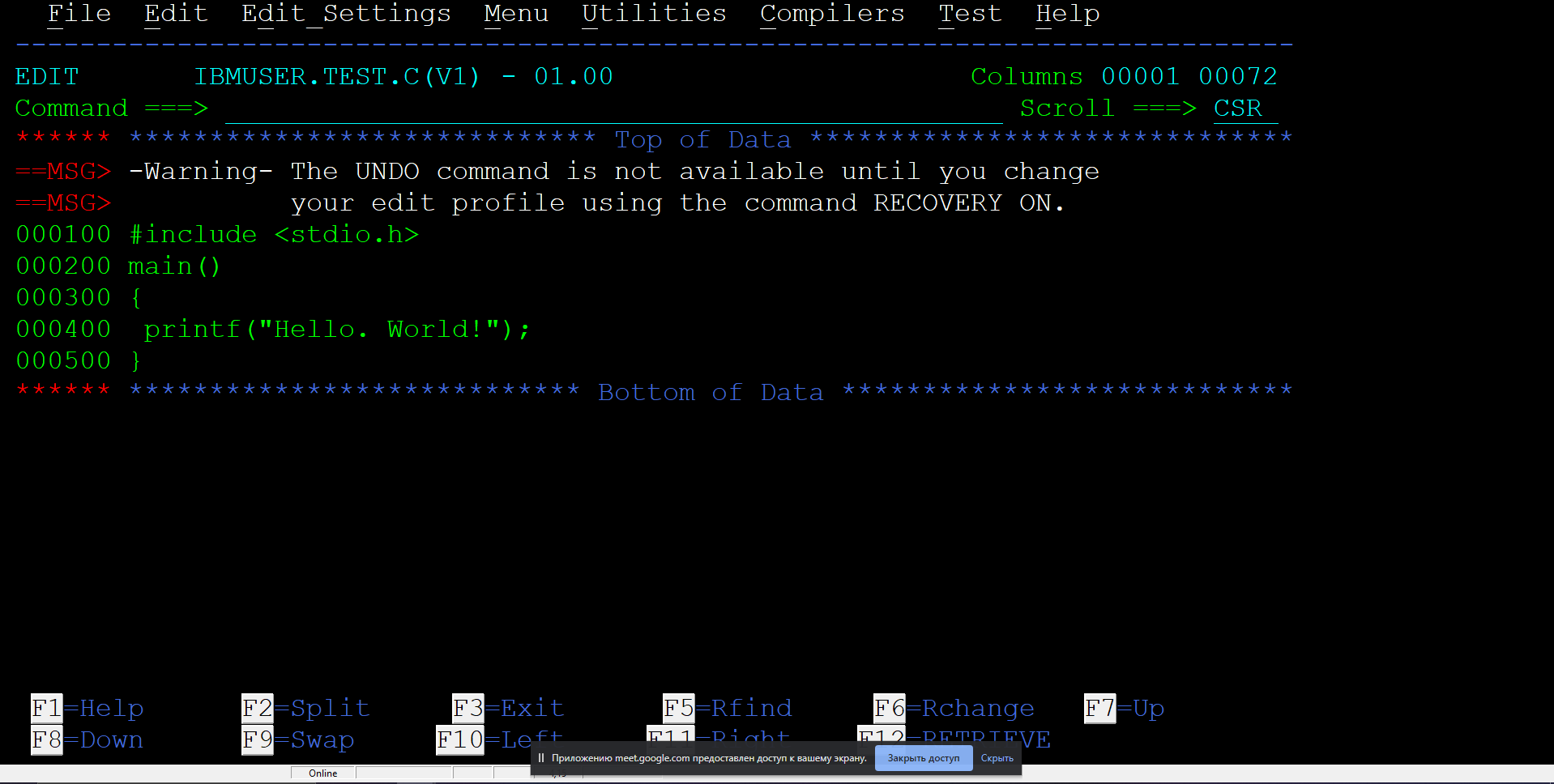


Рисунок 4 – Окно подпрограммы редактирования ISPF с набранной программой на С

Разместите символы точно, как показано. Наша программа предназначена для вывода текстовой строки на экран. Затем нажмите F3 для возврата в главное окно. Программа будет автоматически сохранена.

**Внимание:** Все ключевые слова в коде программы должны быть написаны строчными буквами. Может так случиться, что редактор ISPF автоматически конвертирует эти буквы в заглавные. Если это произойдет, введите в командной строке экрана, показанного на рисунке 4, команду TSO «CAPS OFF», затем нажмите Enter. В правом верхнем углу на экране в окне входа в подпрограмму редактирования ISPF после написания исходных кодов появится надпись о том, что наш файл был сохранен.

**ВАЖНО!** Для освоения функций редактирования в редакторе ISPF вы должны получить у преподавателя учебный курс в электронном виде (на CD-ROMe), изучить его и пройти автоматическое тестирование по этому курсу.

### 4 Создание JCL-скрипта и его выполнение

Пользователь z/OS может запросить у системы выполнение какой-либо работы (конечно, связанной с запуском определенных приложений) с помощью специальным образом записанного и переданного системе текста. Этот текст - задание. Задания составляются на языке управления заданиями **JCL (Job Control Language)** и направляются в систему пользователями через входные устройства и сетевые коммуникации, а также через ранее запущенные приложения. В задании указано, какие программы, в какой последовательности и какими данными должны быть исполнены, а также в какой форме и куда должны быть направлены результаты выполнения программ.

Обычно текст задания создается в некотором наборе данных с помощью текстового редактора, откуда специальными средствами пользовательского интерфейса его можно направить на обработку в подсистему JES2. Система z/OS требует, чтобы для представления заданий использовались только последовательные и библиотечные наборы данных с параметрами FB и длиной записи 80. Создадим задание, отправляющее нашу программу на компиляцию, как файл-раздел (member) в наборе данных формата PDS (Partitioned Data Set) xxx.TEST.СNTL(V1).

Создайте еще один файл-раздел xxx.TEST.CNTL(V1). Введите «CNTL» в поле «Type» и «V1» в поле «Member». Нажмите Enter. Вы попадете в окно редактора ISPF. Введите текст скрипта.

На рисунке 5 – 1, 5, 6 строки ibmuser = ваш user. 1 строка длиною строго в 8 символов, при нехватке добавить цифры в названии скрипта, на 5 и 6 это название файла, не добавлять ничего.

//IBMUSER1 JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,

// TIME=1440

//PROCLIB JCLLIB ORDER=CBC.SCCNPRC

//CCL EXEC PROC=EDCCLG,

// INFILE=' IBMUSER.TEST.C(V1)',

// OUTFILE=' IBMUSER.TEST.LOAD(V1),DISP=SHR'

Замените UserID, который приведен в тексте скрипта на свой. Символ «С» в первой строке скрипта, идущий после UserID без пробела, должен следовать и за вашим UserID, он необходим для идентификации типа работы в планировщике z/OS.

Важно соблюдать расстояния между словами так, как это показано в примере на рисунке 5, т.к. у редактора есть свои «зоны» распознавания исходного текста.

*О структуре скрипта JCL*

Немного слов о структуре скрипта. Задание состоит из последовательности управляющих предложений JCL. Каждое предложение имеет следующую структуру:

//ИМЯ ОПЕРАТОР ОПЕРАНДЫ КОММЕНТАРИЙ

В первых двух позициях всегда указываются две косые черты, которые являются главным отличительным признаком предложений JCL. Поле «имя» начинается с третьей позиции и служит для идентификации представленного в предложении оператора. В нашем примере такие идентификаторы – KC03C0FС, PROCLIB, CCL. Фактически это метки, на которые можно ссылаться из различных предложений задания или других заданий. Имя может содержать не более восьми символов, включающих латинские буквы, цифры и специальные знаки ($ # @) и должно начинаться с буквы или специального знака.

Оператор JOB задает начало задания и режим выполнения задания. Параметр CLASS (класс) относит задание к определенному классу выполнения заданий и задается символами A-Z, 0-9. Параметр MSGCLASS (класс сообщений) определяет выходной класс для системных сообщений, формируемых в процессе выполнения задания. Класс задается символами A-Z, 0-9. Параметр MSGCLASS дает возможность выводить все системные сообщения и выходные наборы данных, формируемые в шагах задания, в один и тот же класс или в разные классы. Такое разделение иногда может оказаться полезным. Параметр MSGLEVEL (уровень полноты сообщений) определяет, какую информацию необходимо выдать в отчет о выполнении задания. MSGLEVEL=(1,1) означает максимально подробный вывод отчета. Параметр TIME (время) устанавливает максимальную продолжительность выполнения задания в минутах.

Оператор JCLLIB задает cписок библиотек для поиска процедур, указанных в задании.

В нашем примере используется библиотека компиляторов CBC.SCCNPRC.

Оператор EXEC указывает выполняемую программу или процедуру и ее параметры. В нашем примере вызывается процедура компиляции программы на языке C – EDCCLG. В ее параметрах – входной набор данных, содержащий текст программы на С и выходной набор данных, в котором будет содержаться полученная после компиляции программа. Параметр DISP (диспозиция) определяет исходное состояние выходного набора данных, а также действия, которые необходимо произвести с набором данных после завершения шага задания или всего задания: сохранить, уничтожить, каталогизировать и др.

Операнд (?) DISP=SHR означает, что набор данных уже существует и может быть использован одновременно другим заданием, т.е. разделяется различными заданиями в режиме чтения.

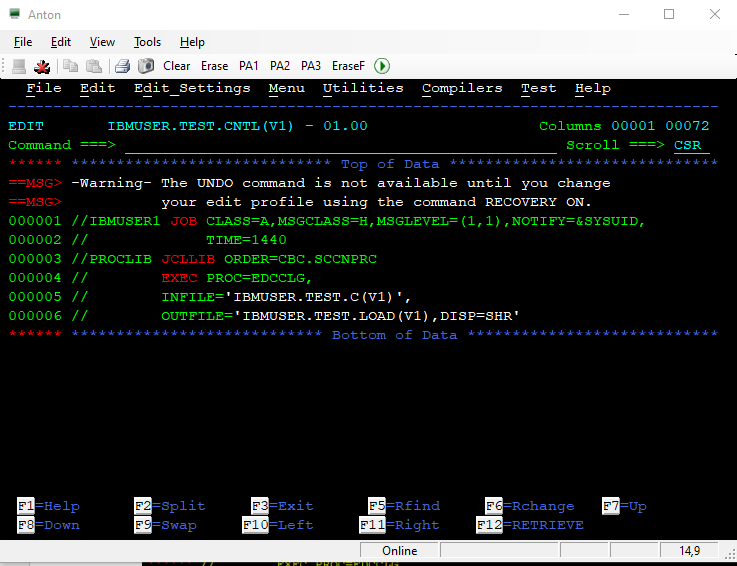


Рисунок 5 – Окно подпрограммы редактирования ISPF с набранным скриптом JCL

При изменении и корректировке файла не забывайте сохранять изменения клавишей F3 (по ее нажатию вы выйдете из редактора, вам нужно будет зайти еще раз).

*Запуск программы на выполнение*

Теперь необходимо отправить задание на исполнение. Задания, поступающие в систему от различных источников, принимаются и обрабатываются специальным компонентом z/OS, который называется подсистемой управления заданиями JES (Job Entry Subsystem). JES принимает задания, поступающие с входных устройств, регистрирует их, осуществляет анализ и формирует очереди заданий, а затем передает задания на выполнение базовой управляющей программе BCP. После завершения выполнения задания и получения результатов от BCP, JES формирует отчет по заданию (листинг), передает его пользователю или выводит на указанные устройства. Для запуска задания на выполнение, находясь в редакторе скрипта, в поле Command введите команду SUB или SUBMIT.

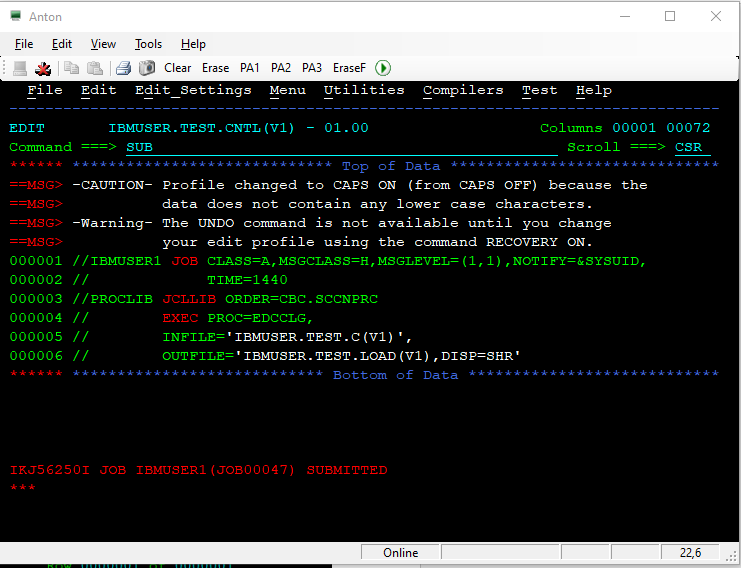
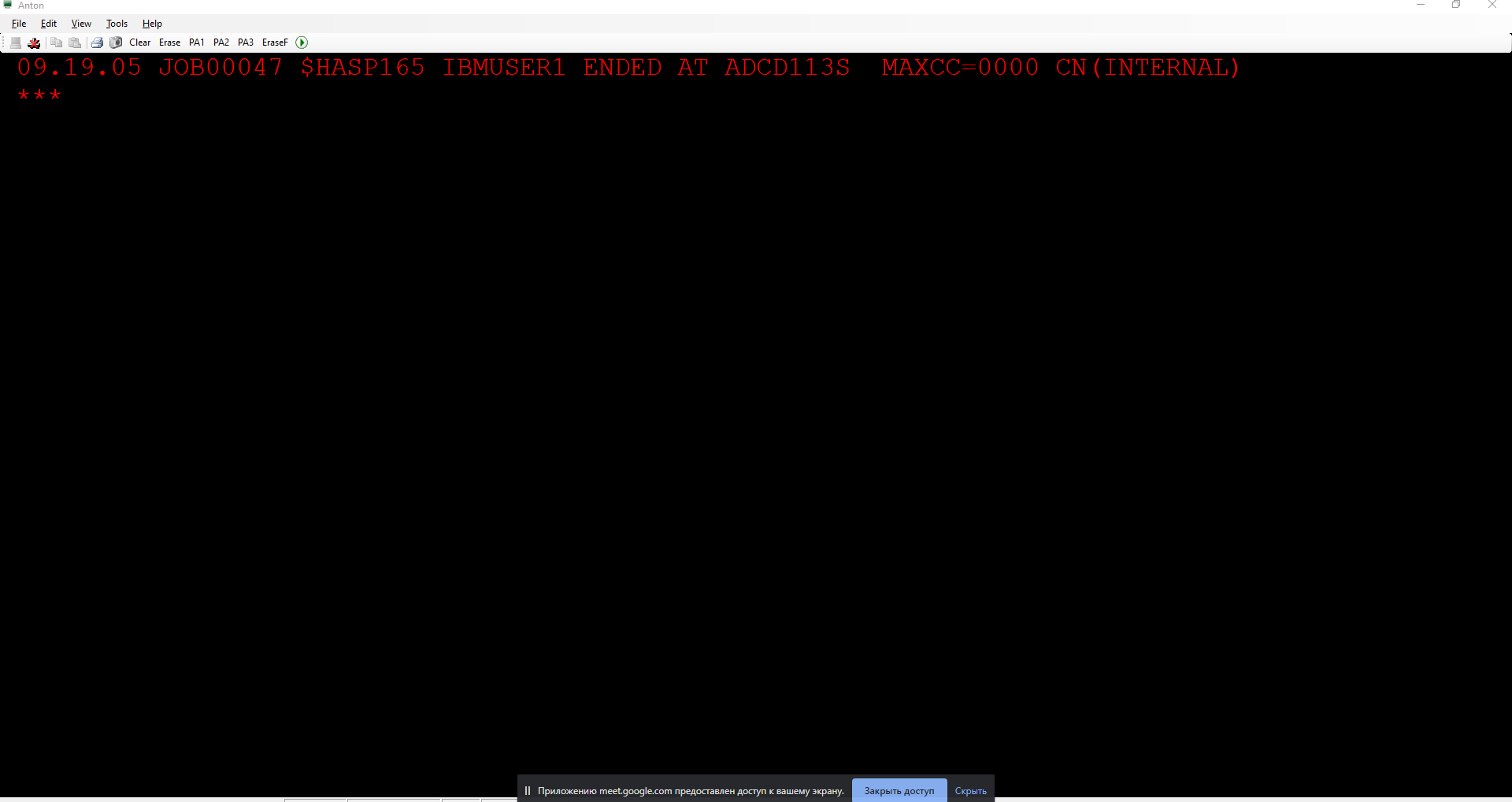


Рисунок 6 – команда SUBMIT

Вы должны увидеть сообщение об отправке задания. Знак «\*\*\*» - значит, что для просмотра остатка сообщения необходимо нажать кнопку Enter. Если скрипт и программа ошибок не содержит, то вы получите сообщение об успехе (MAXCC равен 0).



В противном случае вам придется искать ошибку в скрипте или программе. Для того, чтобы посмотреть отчет об ошибке, можно просмотреть журнал заданий через утилиту SDSF, о чем подробнее будет написано ниже.

### 5 Поиск ошибок и работа с утилитой SDSF

Для того, чтобы посмотреть отчет об ошибке, можно просмотреть журнал заданий через утилиту **SDSF**. Запустить утилиту SDSF можно из среды TSO в интерфейсе под номером 5 (см. рисунок 7).

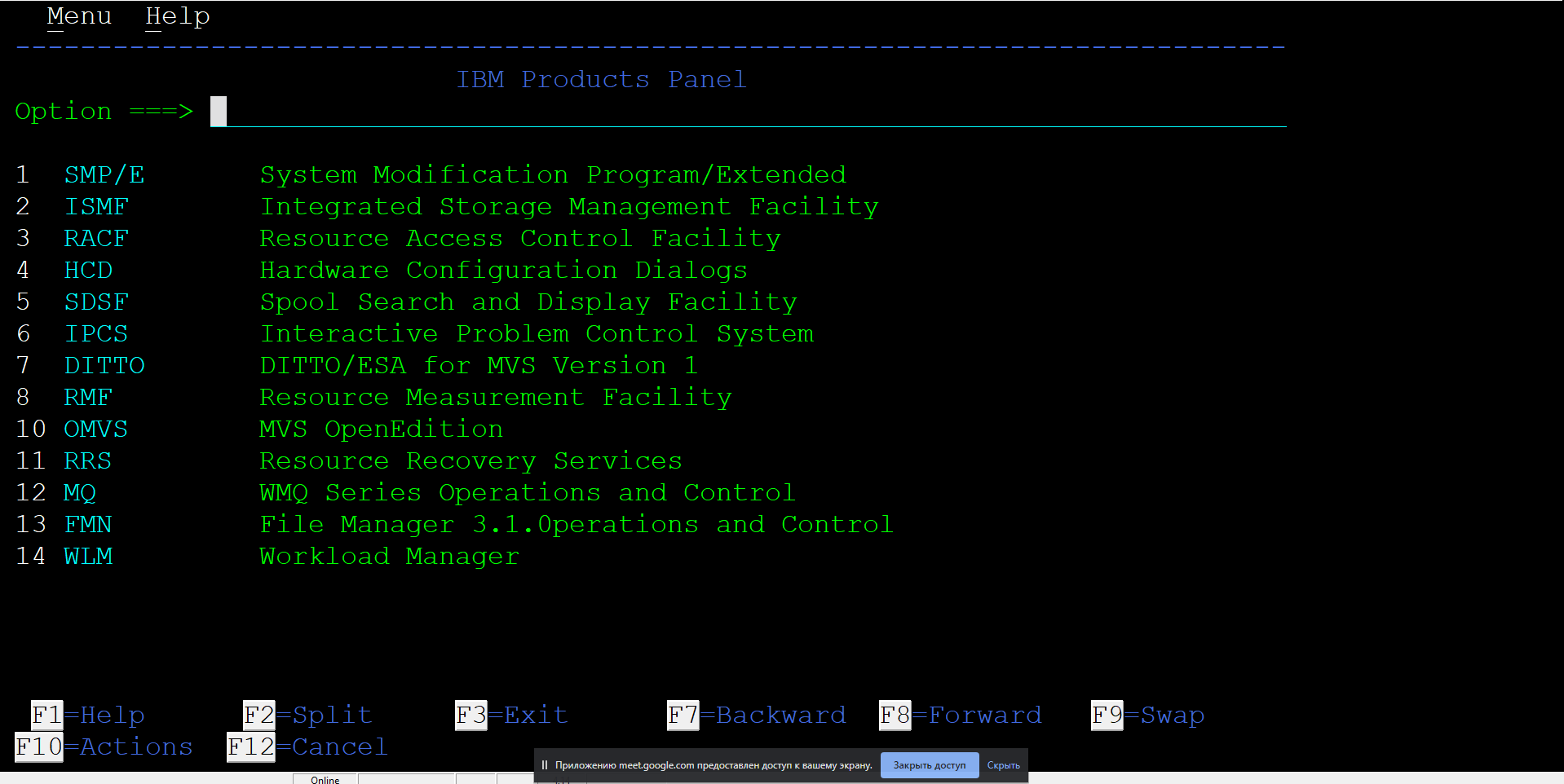


Рисунок 7 – расположение SDSF

Для этого необходимо знать имя задания и UserID. Утилита SDSF выведет список последних запущенных заданий. Установив фильтр, можно быстро найти интересующее нас задание.

Также можно воспользоваться функцией Outlist в ISPF, для этого нужно знать имя задания и идентификационный номер задания. Имя задания задается в первой строке скрипта JCL. Зачастую это UserID и постфикс, в нашем случае - «С», означающий compile – компиляция. Идентификационный номер задания – JobID – можно видеть в отчете об отправке задания, сразу после ввода команды Submit.

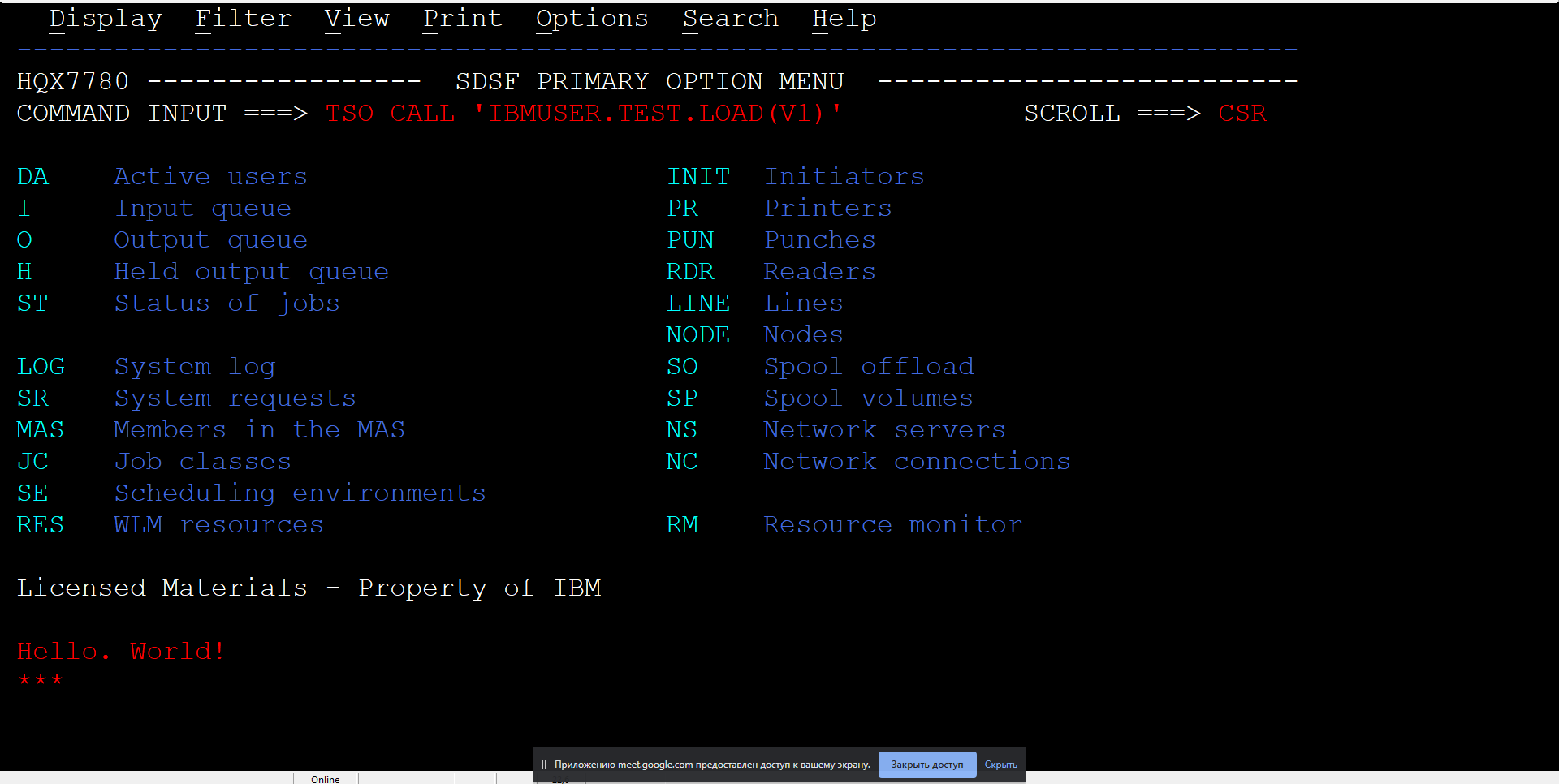


Рисунок 8 – SDSF PRIMARY OPTION MENU

После компиляции двоичный код нашей программы содержится в файле **xxx.TEST.LOAD(V1)**.

Давайте запустим программу. Для этого нам необходимо ввести в полеCommand:

TSO CALL 'KC03C0F.TEST.LOAD(V1)'

Вместо IBMUSER введите свой UserID. После этого вы должны видеть результат работы вашей программы на экране.

### 6 Выход из операционной системы z/OS (z/OS Logoff)

Нажмите несколько раз клавишу F3, чтобы вернуться в основное окно ISPF. Очередное нажатие клавиши F3 приведет вас в окно «**Specify Disposition of Data Set**». Система хочет знать, что вы собираетесь сделать с наборами данных лог-файла, которые вы создали, и предоставляет вам на выбор список действий (рисунок 6). Выберите «3».

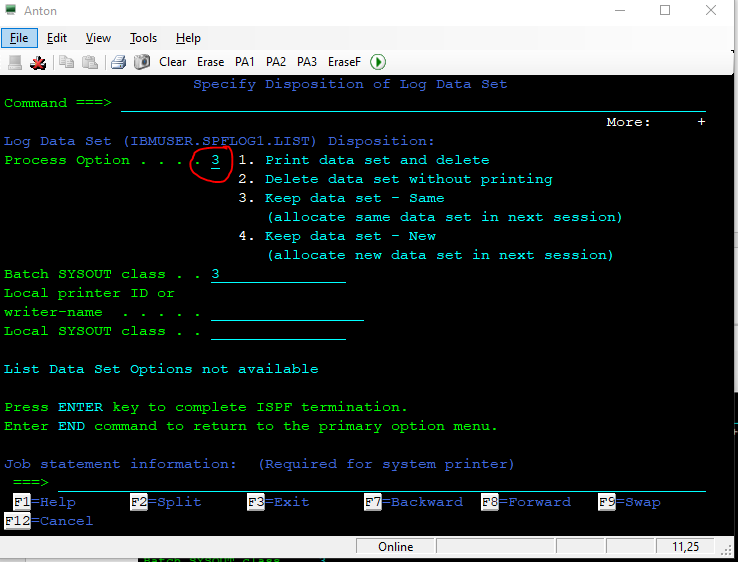


Рисунок 9 – Окно Specify Disposition of Data Set

Сообщение **TSO** о том, что созданный набор данных будет доступен для работы при следующем вашем обращении к системе. При этом набор данных **xxx.SPFLOG1.LIST**, на который ссылается система, был создан ею автоматически.

Затем введите «**LOGOFF**» и нажмите **Enter**.

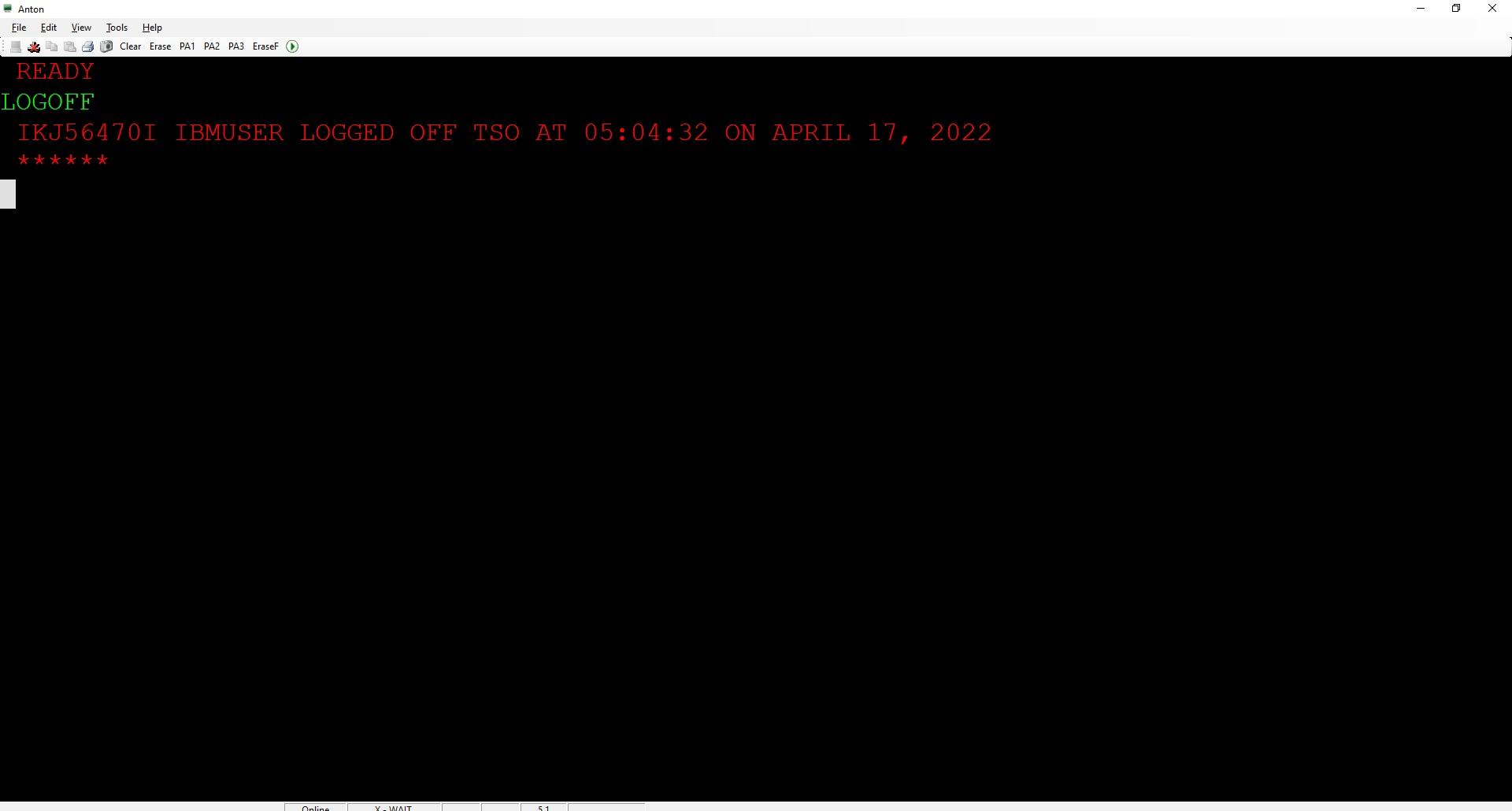


Рисунок 10 – Окно выхода из системы.

Эмулятор 3270 выведет на окно приветствия операционной системы z/OS, ожидающее команд. На этом вы можете закончить сессию связи с виртуальной средой большой вычислительной машины MAINFRAME, выполненное на эмуляторе мейнфрейма Hercules.

### 7 Контрольные вопросы

7.1 Из каких компонентов состояла среда разработки, которую Вы использовали в этой работе? Перечислите инструменты, использовавшиеся в лабораторной работе.

7.2 Опишите процесс создания программы на языке С в среде mainframe?

7.3 Что такое JCL? Что такое JES?

7.4 Каким образом можно скомпилировать готовую программу на языке С в среде mainframe?

7.5 Перечислите основные операторы языка JCL и опишите их назначение.

7.6 Как отправить скрипт JCL на обработку в среду JES?

7.7 Почему тип набора данных xxx.TEST.LOAD был U – undefined?

7.8 Что необходимо сделать, если скрипт JCL вернул ошибку?

7.9 Как запустить скомпилированную программу в среде mainframe?

7.10 В чем разница между компиляцией программы на языке С и программы на другом языке среды mainframe, например COBOL?

7.11 Какие системные библиотеки использовались в JCL скрипте и почему?