



Workshop

What the Hack is the Mainframe?

MLH localhost

IBM Z



1

*Using your Web Browser,
Open this URL:*

<http://mlhlocal.host/lhd-resources>

2

Click on the workshop you're attending, and find:

- Setup Instructions
- The Code Samples
- A demo project
- A Workshop FAQ
- These Workshop Slides
- More Learning Resources



Our Mission is to Empower Hackers.

90,000+
HACKERS

12,000+
PROJECTS CREATED

400+
CITIES

We hope you learn something awesome today!
Find more resources: <http://mlh.io/>

What will you learn today?

1

Who is IBM and what is a Mainframe?

2

Controlling a powerful machine from a Terminal

3

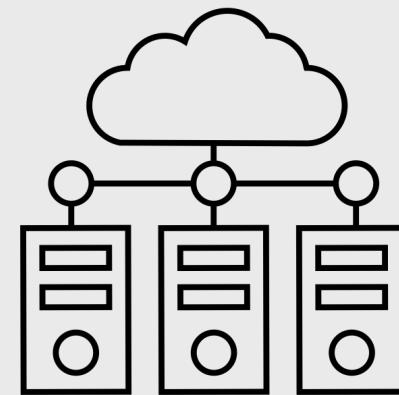
Mainframe file systems and how to navigate them

Table of Contents

- ▶ **0. Introduction to IBM & Mainframes**
- 1. Software setup**
- 2. Connecting to Mainframes & Terminal**
- 3. Navigating the Mainframe**
- 4. z/OS File systems**
- 5. Review!**

What is a Mainframe?

A Mainframe is a very large and powerful computer, capable of running extremely complex & demanding applications for thousands of users simultaneously.

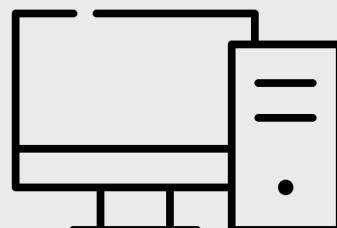


The mainframe offers **99.999%** availability, often referred to as “Five Nines”, while processing **1.2 million transactions per second**

What is a Mainframe?

A mainframe is a type of computer that is specifically built for high-volume workloads, full-time availability, and full data encryption. They serve businesses around the world 24/7.

In this workshop, you'll connect to a mainframe and hack around with a few simple challenges.



What is a Mainframe?

The mainframe is meant to be accessed remotely, which is what makes it possible for 1000's of people to access these machines at the same time.

There are protocols designed specifically for this practise.

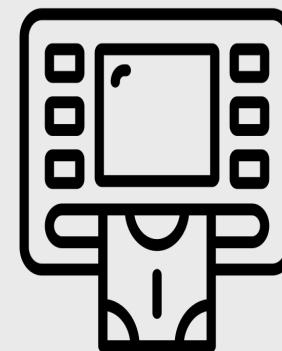
Here are some well known ones.

- Web
- Terminals
- APIs

Who uses Mainframes?

Everyone!

If you have ever used a credit card or withdrawn money from an ATM, a Mainframe was behind that transaction.



Companies that run demanding and popular web applications that require simultaneous access to the same data benefit from using Mainframes.

Who is IBM?

IBM is one of the largest hardware & software companies in the world.

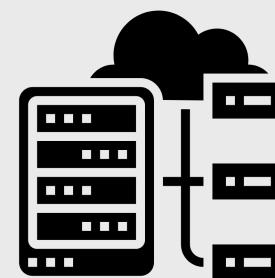
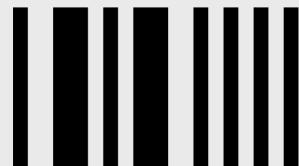


They have been designing, engineering, and building solutions in software, hardware, and technology for over a century.

You might recognize some of their significant contributions

IBM is responsible for ...

- Mainframe computers
- Deep Blue Chess Ai
- Quantum Computers
- ATM Machines
- Barcodes / Universal Product Codes
- Relational Databases



And so much more.. seriously!



MASTER THE MAINFRAME

BE A PART OF ONE OF THE WORLD'S
LARGEST STUDENT COMPETITIONS!

SEPTEMBER 9th - DECEMBER 31st 2019
Register Now!

Cool! That's a little about IBM

The next steps are going to guide you through

- Downloading & running a 3270 emulator
 - Logging into the Mainframe
 - Changing your account password
- Learning to navigate and use the Mainframe!

Table of Contents

- 0. Introduction to IBM & Mainframes**
- 1. Software setup**
- 2. Connecting to Mainframes & Terminal**
- 3. Navigating the Mainframe**
- 4. z/OS File systems**
- 5. Review!**

So, what's this 3270?

- Mainframes can accessed remotely by dedicated systems called Terminals. 3270 refers to the family of terminals that were used to access these mainframes.
- Today, it is possible to emulate these systems entirely in software, which is why the software is called a 3270 Terminal *Emulator*.

There are many ways to connect to a mainframe, and some a lot of people prefer using 3270!

Access with 3270

3270 Terminal sessions allow for reliable connections while not requiring a lot of network bandwidth

However not every computer provides a 3270 utility, so you will need to install software known as an **emulator**.

Software Requirements

You will need to download an emulator from one of the URL's provided.

mlhlocal.host/3270-Windows

mlhlocal.host/3270-Mac

Select the URL provided for your operating system!

Software Requirements

Windows, Mac and **Linux** are all supported slightly different to one and other.

So you have different downloads to make 3270 accessible on your computer.

Don't worry! We'll take you through every step of that and help you set up!

Access with 3270: Mac

For the Mac connection we're going to use a free application known as [tn3270](#).

1. On the provided website, select [HTTP download](#).
2. You will download a disk image(dmg) file. Click [HTTP download](#) to open up the tn3270 application!

Download the Latest Version of tn3270

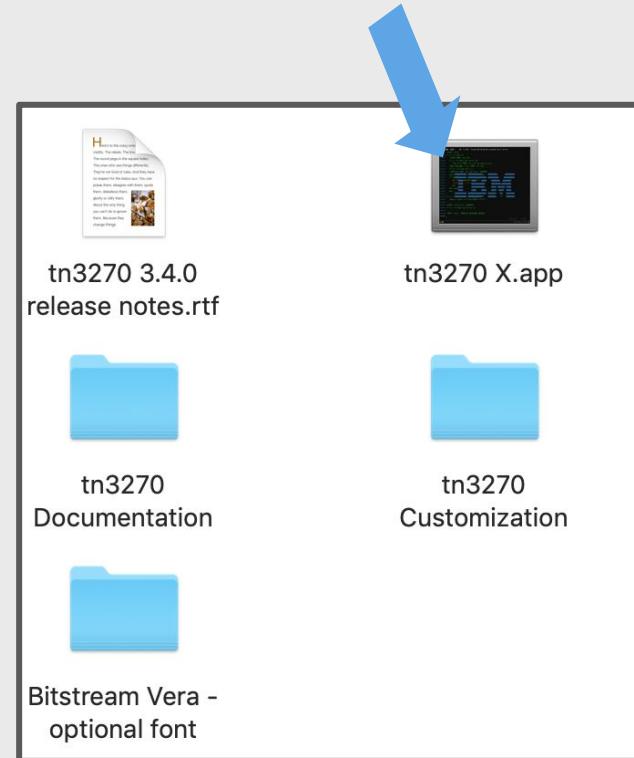
Version 3.4.0 is an update to version 3.3 that adds support for latest versions of OS X, it also includes experimental support for OS X 10.9 and later.

- OS X Version 3.4.0, November 9, 2013, 2.0 MB (for OS X 10.3.9 and later)
[HTTP download](#)
[Alternate site download](#)

 tn3270_X_3.4.....dmg

Access with 3270: Mac

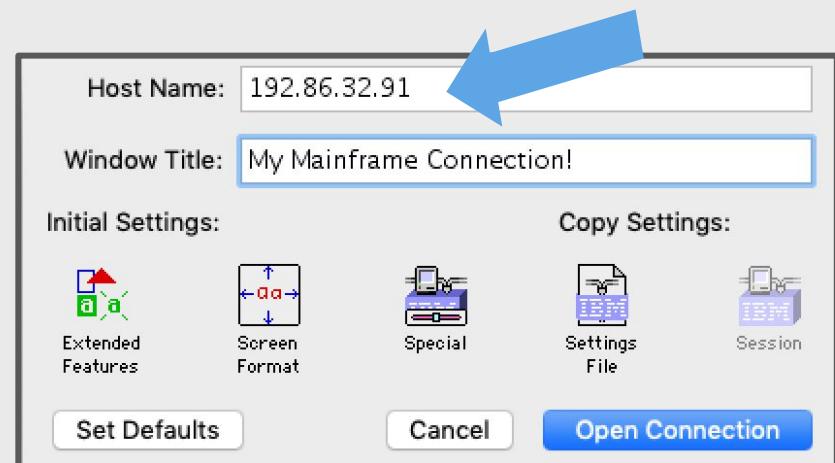
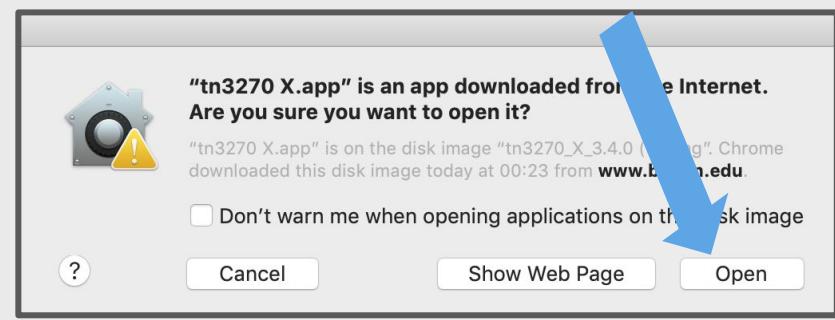
3. Double-Click **tn3270 X.app** to run your 3270 emulator.



The Mac download doesn't require a long installation process, once the disk image is open you can run the app locally!

Access with 3270: Mac

3. Click open to access the application.
4. You should now be looking at the User Interface which asks you for a **host name** otherwise known as an *internet facing IP address*. Enter this IP Address for the host name **192.86.32.91**



Access with 3270: Windows

1. Navigate to the webpage provided for the Windows telnet emulator [Vista TN3270](#) download.

The screenshot shows a website for 'Tom Brennan Software'. The header features a logo of a computer monitor with a stack of books, followed by the text 'Tom Brennan Software' in blue script. The main content area is titled 'Home of the **Vista tn3270** Terminal Emulator'. It includes a red checkmark icon and text about the product being a Windows program for emulating IBM 3270 terminals via IP link, available for a free 30-day trial at \$30. It highlights unique features compared to commercial emulators. A sidebar on the left lists links: Home, Features, Screen Shots, Requirements, Download, Ordering, Support, Vista Blog, About Me, and Other Items. At the bottom, there are sections for 'Features' and 'Screens'.

Vista tn3270 is a Windows program designed to emulate IBM 3270 terminals connected to a host via IP link. Currently it is available for a free 30 day trial, and costs only **\$30**. If you are looking for an emulator created with **mainframe programmers** in mind, then give this one a try. You might find some unique features unavailable even on the highest priced commercial emulators.

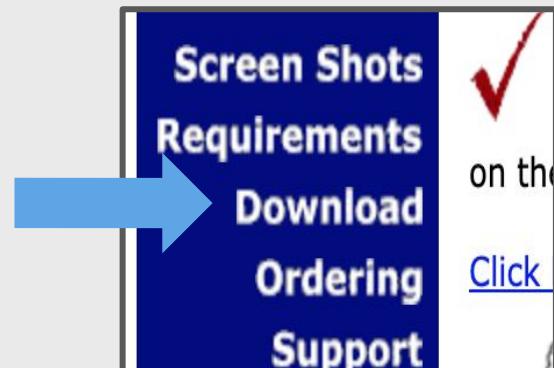
[Click here to download.](#)

Features Vista has features designed especially for programmers, such as built-in multiple cut and paste buffers, fully tailorable keyboard, extensive select/copy/paste

Screens Vista uses bitmapped raster fonts for the clearest text possible. There are 2 sets, "Thick" and "Thin", in 73 sizes each from 4x6 to 16x36. With so many sizes you

Access with 3270: Windows

2. Navigate to **Download** in the left side menu.



3. Click the download link and you'll find received an exe file. Run this to install your new emulator.

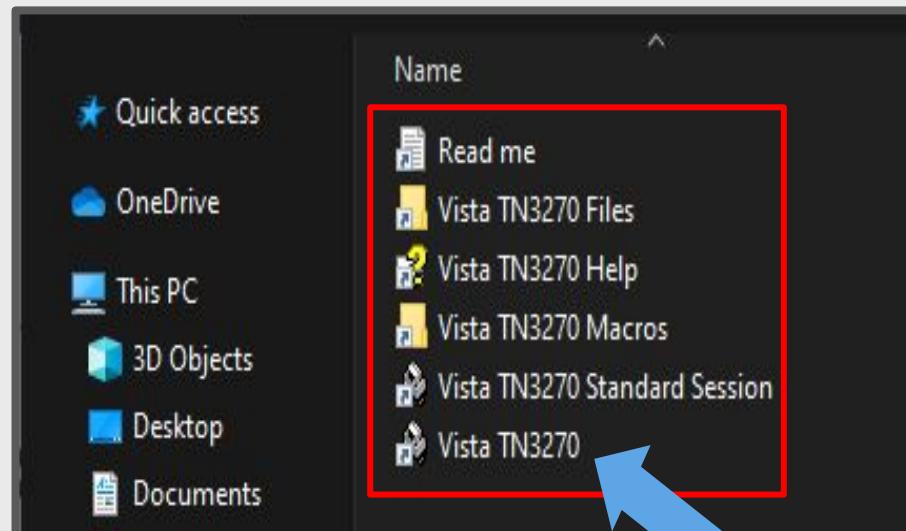
4. Follow the installation instructions.



Access with 3270: Windows

You will see a few shortcuts have been created with the installation.

5. The installation will generate several shortcuts - you are going to need [Vista TN3270](#), and [Vista TN3270 Standard](#).



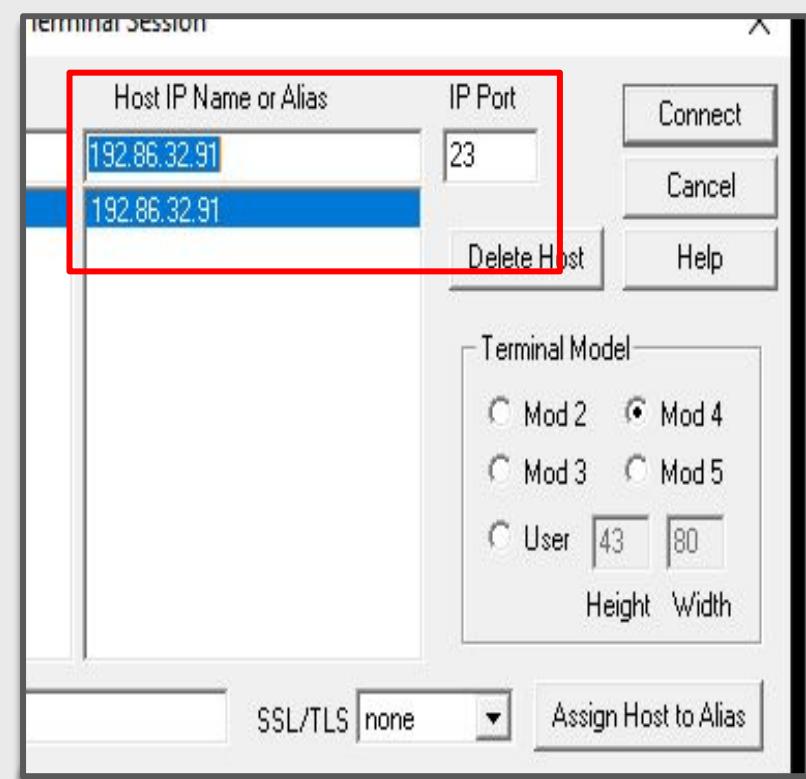
6. Let's move on by clicking & opening [Vista TN3270](#).

Access with 3270: Windows

This will prompt a new terminal session.

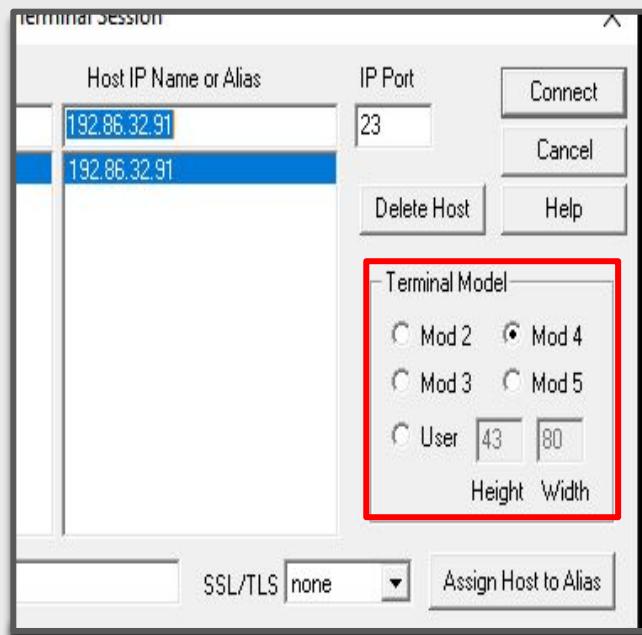
7. Enter the provided IP for your Mainframe if the field **"Host IP Name or Alias"**. Enter this IP Address for the host name **192.86.32.91**

8. The default port to use is **23**. Make sure your **IP Port** field contains **23**.



Access with 3270: Windows

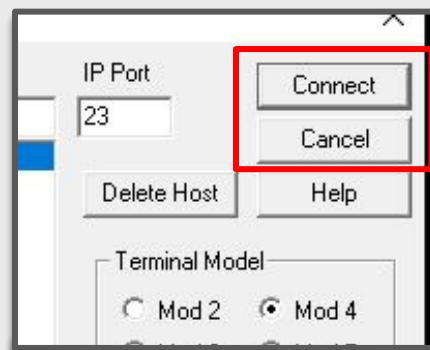
9. **Terminal Model** represents the size of the terminal screen when you connect. Selecting Mod 4 will create a nice, visible interface for you.



You can select a custom terminal **Height** and **Width** if you have a preference.

Access with 3270: Windows

10. Click **Connect**. Once you click connect the emulator will load and will connect to an IBM Mainframe.



Welcome to z/OS!



*Time to become the Mainframe Guru you
always wanted to be!*

Awesome! You're connected to the Mainframe!

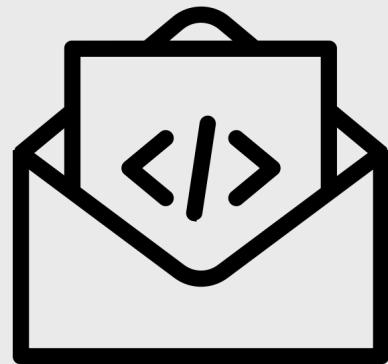
Now the connection is established, what's next?

Now we will

1. Successfully log in
2. Change passwords
3. Explore the Z/OS file system!

z/OS Credentials

In order to complete this workshop you need to Register for the **Master the Mainframe Contest** in order to get your z/OS credentials!



The credentials needed to access the z/OS mainframe are the [Username](#) and [Password](#). Which will be provided to you via email after signing up.

z/OS Credentials

To register for the contest. Go to the following link and complete the sign-up.

mlhlocal.host/master-the-mainframe



z/OS Credentials

IBM Master the Mainframe Registration

Please complete the following registration form

The fields indicated with an asterisk (*) are required to complete this transaction; other fields are optional. If you do not want to provide us with the required information, please use the "Back" button on your browser to return to the previous page, or close the window or browser session that is displaying this page.

Information

First Name*:

Last Name*:

E-mail Address*:

Confirm E-mail Address*:

Country*:

Street Address*:

City*:

Postal Code*:

School*:

Social Media Profile:

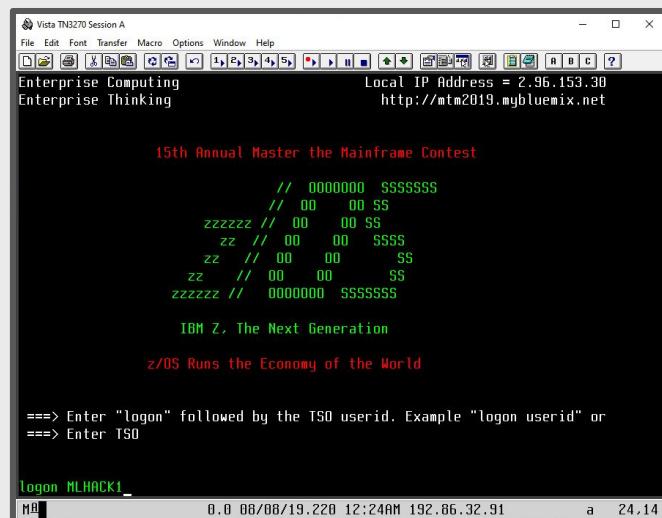
Once you complete the registration process, you will receive an email within 15 minutes with your z/OS credentials that will allow you to access the mainframe!

Go back to your 3270 Emulator and let's explore the interface.

3270 Interface

First! Let's get to know what you are now looking at.

You just connected to a remote machine, with a new tool you downloaded. What is going on!



This is the user interface for the z/OS Operating System running on an IBM Mainframe.

It's a text based interface, that you control entirely by typing commands.

3270 Interface

What you are seeing isn't a greeting screen.
This **IS** the z/OS interface!

Let's get better acquainted and log in!



At the bottom, you are able to type into the interface.

Here you are able to enter the logon command to proceed.

Logging in

1. Type **logon** followed by your provided username!
2. Hit the **ENTER** key to submit.

```
==> Enter "logon" followed by the
==> Enter TSO

logon MLHACK1_
MA| 0.0 08/08/19
```

Notice: Hitting the backspace does not delete your previously typed character. It only moves your flashing caret back 1 space. To clear a character, you need to enter **SPACE** when the flashing caret has moved onto the character you need to remove.

Resetting and Reconnecting

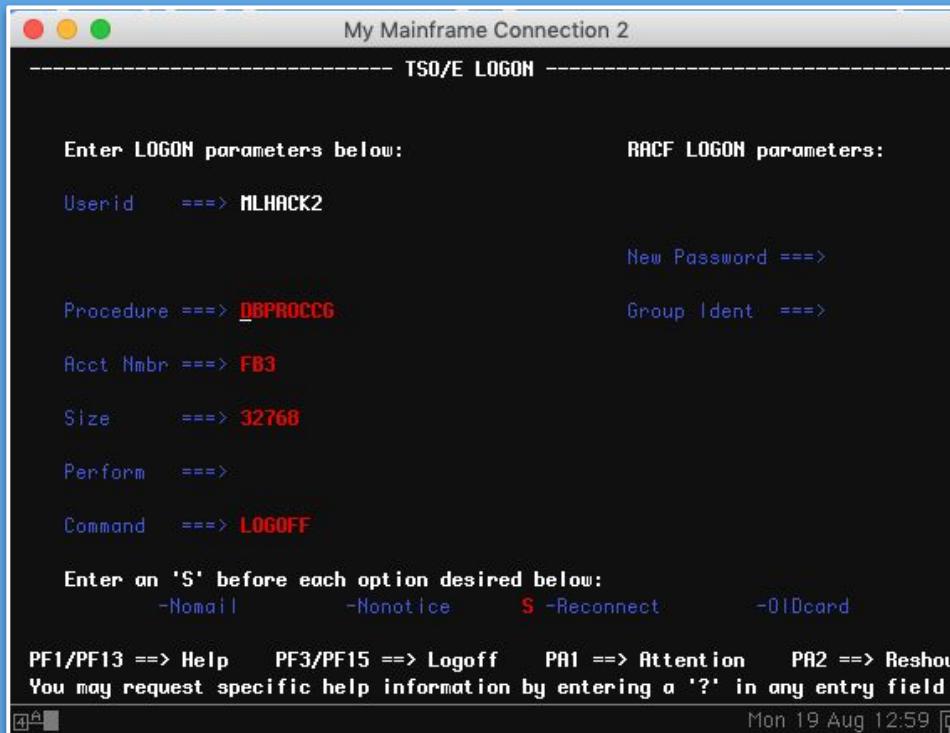
If you make a mistake, don't worry! You are able to completely restart your session by clicking the **Reconnect** button in the emulator toolbar, or just close the emulator and restart



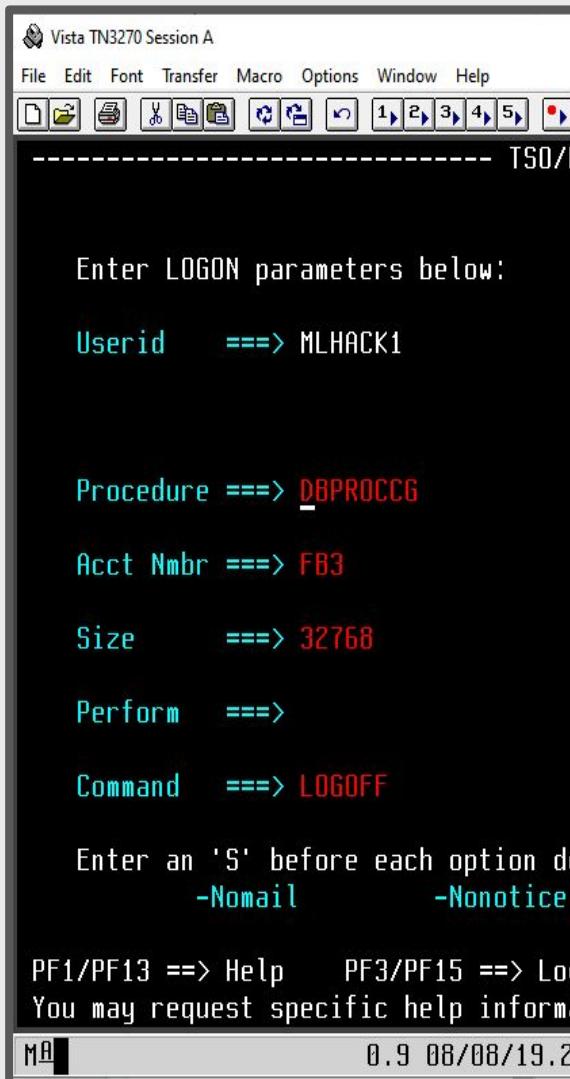
You logged into the Mainframe!

Now you're looking at the interface of your very own user environment, running on an **IBM Mainframe**.

Let's go over what you're looking at.

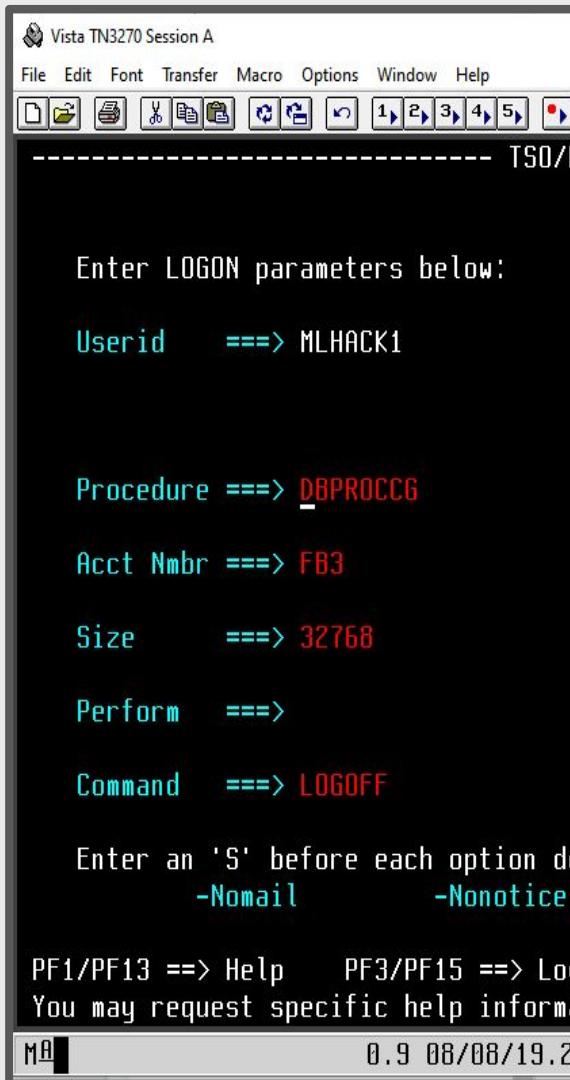


On the left



- The **Userid** field displays the User ID for logging in.
- The **Procedure** field displays the method used to log on the Mainframe. (If you were to configure your own mainframe you would choose, or write a procedure for this).
- The **Size** indicated the amount of space that will be allocated to you in your isolated time sharing environment (in KiloBytes/KB).

On the left



- The **Perform** field indicates your Performance Group, which level of administrator access you have.
- The **Command** field indicates what command will be executed **AFTER** z/OS is finished running all administrator specified commands.

On the right

- The **New Password** field allows you to type in and confirm a new password for your TSO User ID.
- The **Group Ident** field could specify your security access group, if you had one.



At the bottom

```
Command ==> LOGOFF
```

Enter an 'S' before each option desired below:

-Nomail

-Nonotice

-Reconnect

-OIDcard

PF1/PF13 ==> Help PF3/PF15 ==> Logoff PA1 ==> Attention PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field

- **-Nomail** and **-Nonotice** allow users to enable/disable personal and group-wide notice messages that the Mainframe Time Sharer display.
- **-OIDcard** is used to perform operations from a magnetic ID card through a physical reader. (Cool!)

Reconnecting to the mainframe

```
Command ==> LOGOFF
```

Enter an 'S' before each option desired below:

-Nomail -Nonotice **-Reconnect** -OIDcard

PF1/PF13 ==> Help PF3/PF15 ==> Logoff PA1 ==> Attention PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field

-Reconnect is important. This command lets users re-establish connections to the Mainframe incase of an accidental disconnection and were unable to log out completely. The TSO may refuse to log you in until your session ends.

Reconnecting to the mainframe

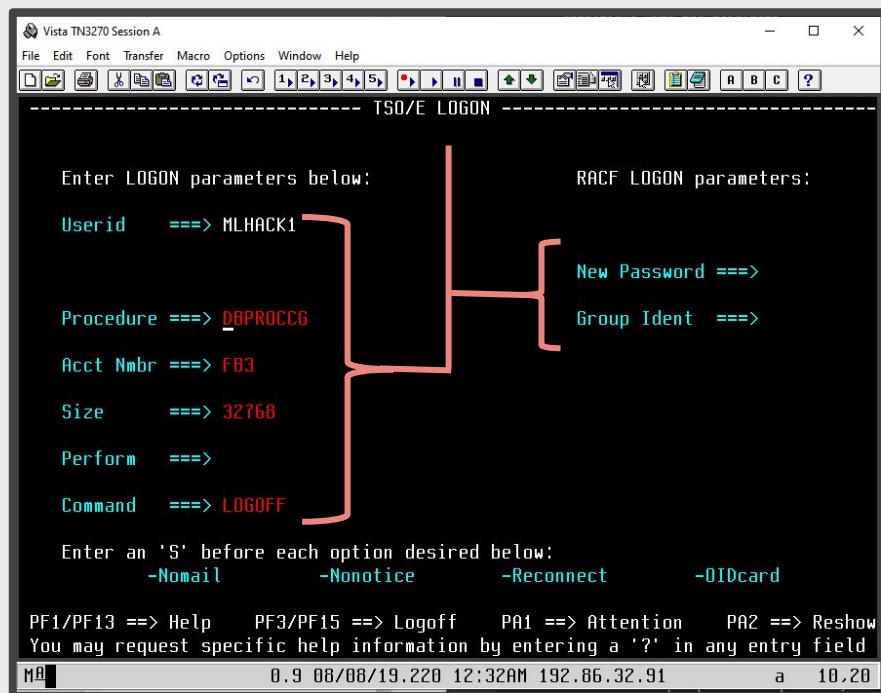
You activate this by navigating to -Reconnection, typing **S** and pressing **Enter**.

```
desired below:  
ce      S -Reconnect      - -0  
logoff  PA1 = Attention  
motion by entering '2' in a
```



Navigating the Mainframe

Navigation on a Mainframe with 3270 is done by using **arrow keys** and/or **TAB** and then selecting options using **ENTER**, or by typing directly on the user interface.



So let's begin!

1. Move your caret around using the arrow keys **up, down, left and right**. However, to avoid errors it's **highly recommended** to use the **TAB** key to traverse whole sections of the interface.

Changing your Password

As a first task, Let's learn to manually change your password.

Spend a moment coming up with a new password for your account.

It must:

- *Be equal to or less than 8 characters*
- *contain at least one letter*
- *Contain at least one number.*
- *not be the same as a previously z/OS password.*

Manual password reset

To use the **New Password** ability, simply hit **ENTER** and you'll be prompted to type the new password you wish to use.

Any text input will be stored as a new password.

You will be asked to input the password twice. Then you will have successfully changed passwords.

As you type your password, no characters will appear. This is to maintain security. So take care with what you type.

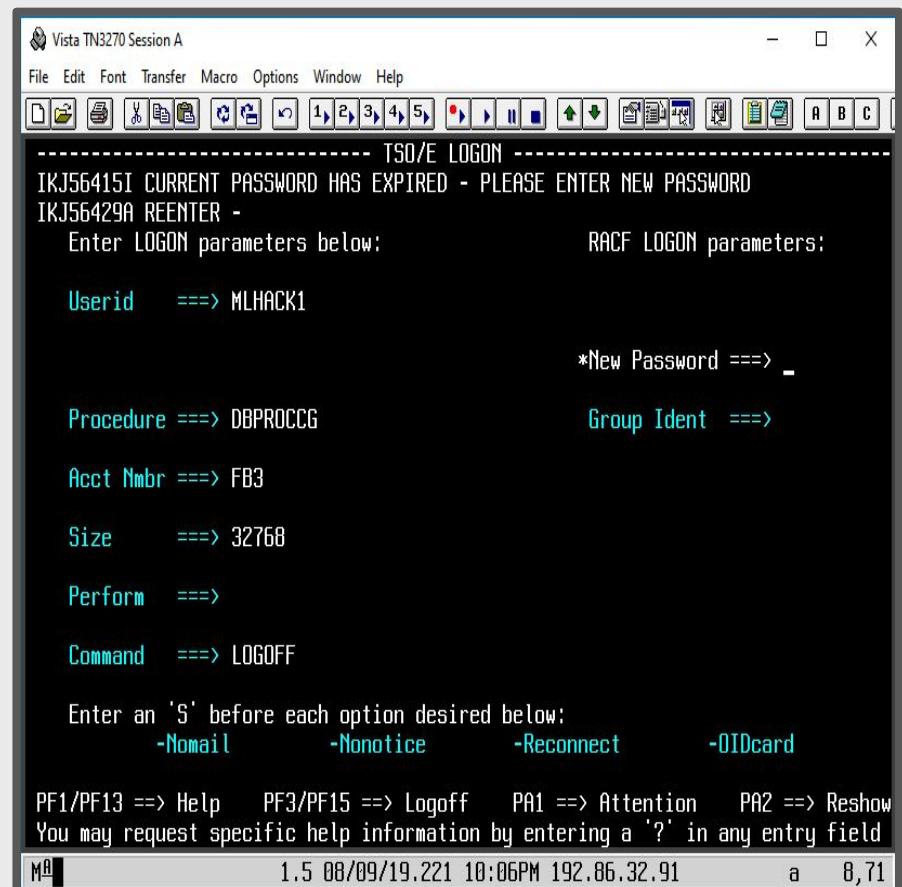
```
RACF LOGON parameters:  
*New Password ==> -  
Group Ident ==>
```

Manual password reset

If at any stage you input your password incorrectly.

DON'T PANIC.

You can simply repeat the process until you successfully change the password for your account.



Navigation Basics? Check!

You're doing great!

Logging on and navigating the ISPF is a great achievement and the first in many steps to mastering the art of working an IBM Mainframe.

Let's move onto TSO, short for Time Sharing Option. This is what allows for multiple users to access the system at once without slowing things down.

z/OS Time Sharing Option

So what is Time Sharing?

Time Sharing Option (TSO) is a dedicated environment allocated for you, in real time, on the z/OS system.

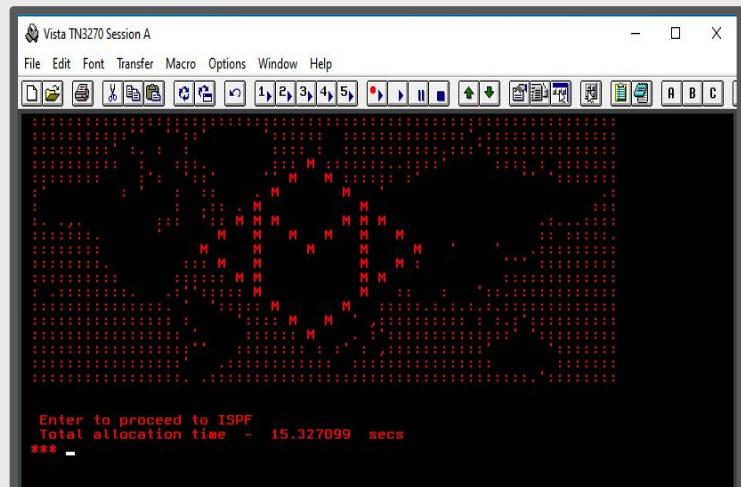
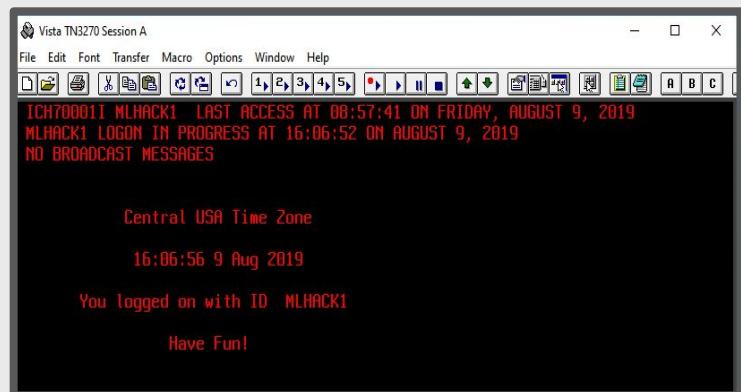
This allows you to get access to resources, and use them as if you have full dedicated access, even though they are being used by others at the same time.

It's essentially your own private computer session to run your applications or code!

Time Sharing Option

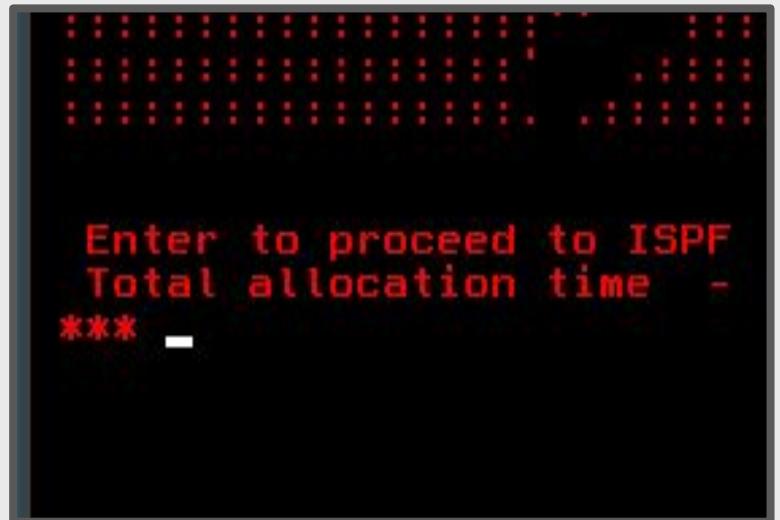
- Once you enter a new password successfully you will automatically be logged on to TSO.
- This will take approximately 10 seconds. It will feel like the system has stalled, but it hasn't.

Don't worry!



Time Sharing Option

To access the **ISPF** menu from the **TSO** screen we don't need any direct input. Just the **ENTER** key.



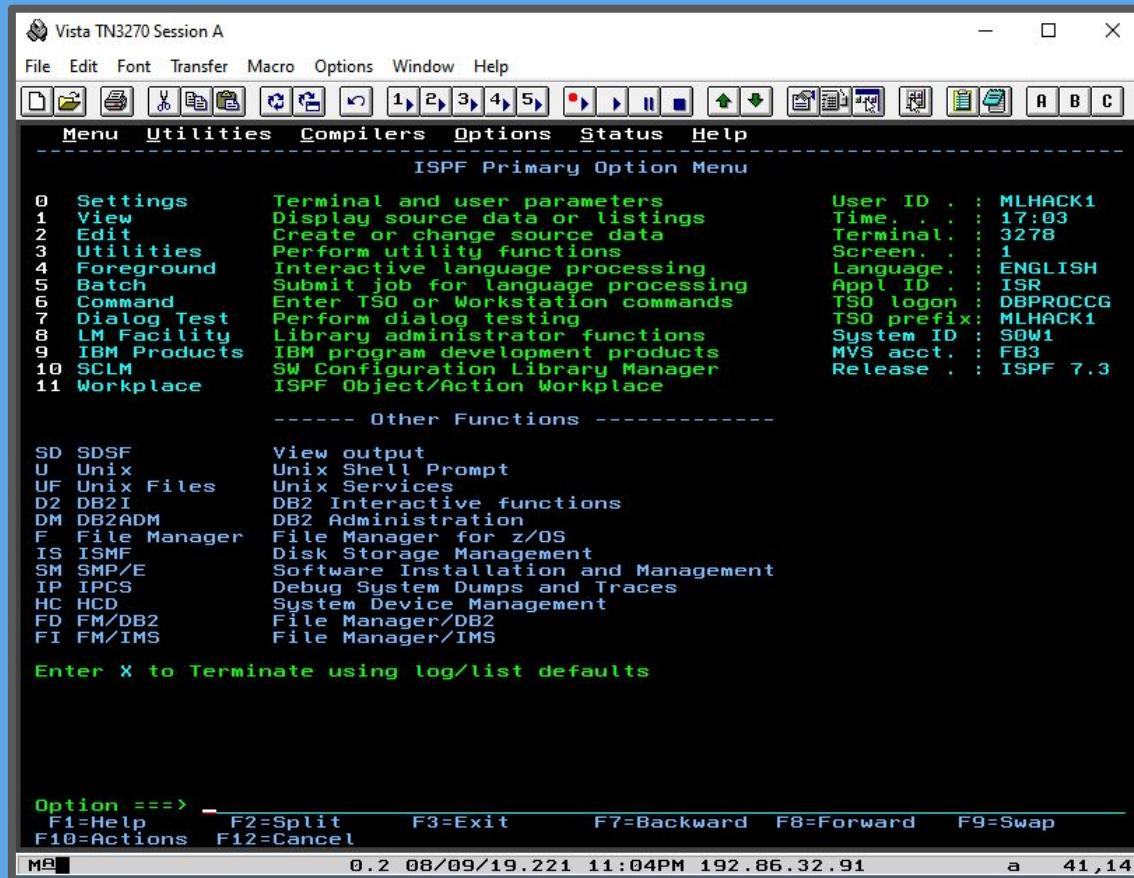
Tip



The “***” is a prompt for user input. It means the screen is being held until you hit Enter to continue.

Enjoy this ISPF

Take a breather!



ISPF

These are the primary ISPF options. We will spend most of our time manipulating the Mainframe with these.

```
0 Settings      Terminal and user parameters          User ID . : MLHACK1
1 View          Display source data or listings        Time. . . : 16:08
2 Edit          Create or change source data         Terminal. : 3278
3 Utilities     Perform utility functions           Screen. . : 1
4 Foreground    Interactive language processing       Language. : ENGLISH
5 Batch          Submit job for language processing   Appl ID . : ISR
6 Command       Enter TSO or Workstation commands     TSO logon : DBPROCCG
7 Dialog Test   Perform dialog testing              TSO prefix: MLHACK1
8 LM Facility   Library administrator functions     System ID : SOW1
9 IBM Products  IBM program development products   MVS acct. : FB3
10 SCLM         SW Configuration Library Manager   Release . : ISPF 7.3
11 Workplace    ISPF Object/Action Workplace
```

ISPF stands for

Interactive System Productivity Facility

It is the core of how users like you, control Mainframes.

Key Term

ISPF: Stands for Interactive System Productivity Facility, is a full panel application navigated by keyboard which includes a text editor, browser, and functions for locating and listing files and performing other utility functions.

Let's Explore

To get a better understanding of this system and its total usage, let's take a look at [SDSF](#).

```
----- Other Functions -----
SD SDSF          View output
U Unix           Unix Shell Prompt
UF Unix Files   Unix Services
D2 DB2I          DB2 Interactive functions
DM DB2ADM        DB2 Administration
F File Manager   File Manager for z/OS
IS ISMF          Disk Storage Management
SM SMP/E         Software Installation and Management
IP IPCS          Debug System Dumps and Traces
HC HCD           System Device Management
FD FM/DB2         File Manager/DB2
FI FM/IMS        File Manager/IMS
Enter X to Terminate using log/list defaults
```

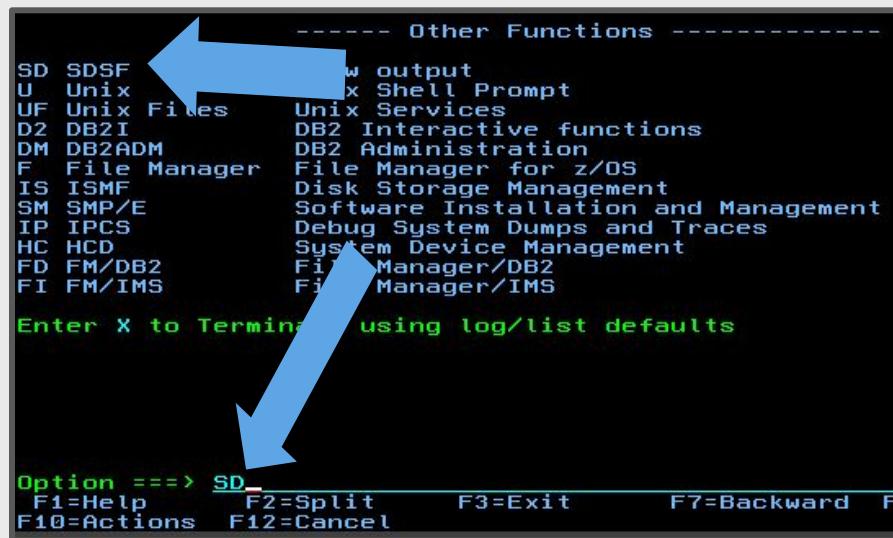
This stands for [System Display and Search Facility](#), and it is one of the supplementary options in [ISPF](#).

Key Term

SDSF: Stands for System Display and Search Facility, is a utility that allows you to monitor, control, and view the output of jobs in the system.

Let's Explore

1. To enter SDSF, type the command **SD**.
2. Press **ENTER**.



SD is described at the top of the '**Other Functions**' list. It will allow you to View the Output of events taking place on the system.

Let's Explore

You will be presented with a system management screen with a lot of new options.

The commands are in the *Menu column*, the command descriptions are in *Description column*.

To execute these commands, you type in the command name and hit **ENTER**.



Tip



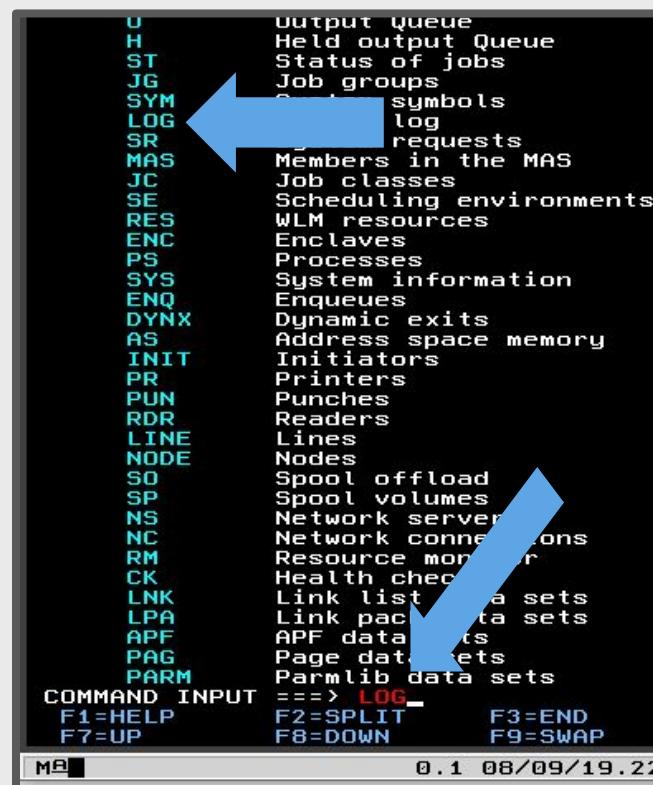
To go back from any current position, press **F3**.



Let's Explore

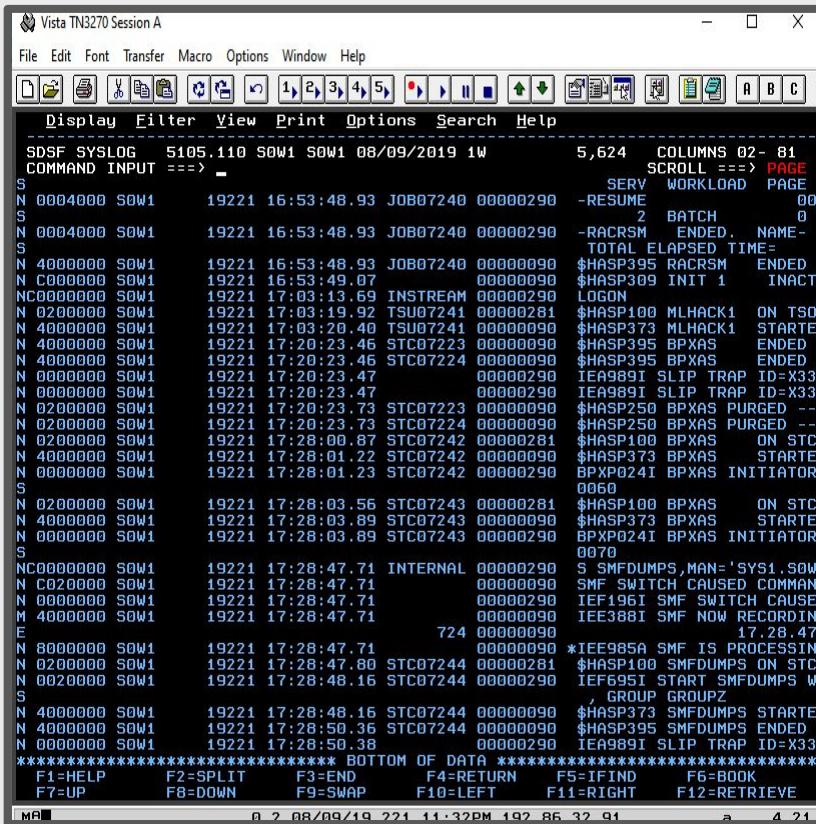
1. Type **LOG** and hit **ENTER**.
2. Begin to look around! See if you recognise any events listed. Use the **F7** & **F8** keys to scroll up and down the logs. Use the **F10** & **F11** keys to scroll up and down.

Since you have been poking around the Mainframe for a while now. Let's look inside the system logs. Perhaps you have generated some events that have been recorded.



```
Output queue
H Held output Queue
ST Status of jobs
JG Job groups
SYM Symbols
LOG Log requests
SR Requests
MAS Members in the MAS
JC Job classes
SE Scheduling environments
RES WLM resources
ENC Enclaves
PS Processes
SYS System information
ENQ Enqueues
DYNX Dynamic exits
AS Address space memory
INIT Initiators
PR Printers
PUN Punches
RDR Readers
LINE Lines
NODE Nodes
SO Spool offload
SP Spool volumes
NS Network server
NC Network connections
RM Resource monitor
CK Health checker
LNK Link list data sets
LPA Link pack data sets
APF APF data sets
PAG Page data sets
PARM Parmlib data sets
COMMAND INPUT ==> LOG
F1=HELP F2=SPLIT F3=END
F7=UP F8=DOWN F9=SWAP
0.1 08/09/19.22
```

Let's Explore



The screenshot shows a Windows Vista TN3270 session window titled "Vista TN3270 Session A". The window displays a log of system activity from a mainframe. The log includes columns for timestamp, user ID, job name, and various system events like RESUME, RACRSM, and SMF dumps. The log ends with a note about reaching the bottom of the data.

Time	User	Job	Event	Details	
19221 16:53:48.93	N 0004000 SOW1	JOB07240	-RESUME	SERV WORKLOAD PAGE 00	
19221 16:53:48.93	N 0004000 SOW1	JOB07240	-RACRSM	ENDED, NAME- 2 BATCH 0	
19221 16:53:48.93	N 4000000 SOW1	JOB07240	TOTAL ELAPSED TIME=	\$HASP395 RACRSM ENDED	
19221 16:53:49.07	N C0000000 SOW1			\$HASP309 INIT 1 INACT	
19221 17:03:13.69	N C0000000 SOW1	INSTREAM	LOGON	\$HASP373 MLHACK1 ON TSO	
19221 17:03:19.92	N 0200000 SOW1	TSU07241	000000281	\$HASP100 MLHACK1 ON TSO	
19221 17:03:20.40	N 4000000 SOW1	TSU07241	00000090	\$HASP373 MLHACK1 STARTED	
19221 17:20:23.46	N 4000000 SOW1	STC07223	00000090	\$HASP395 BPXAS ENDED	
19221 17:20:23.46	N 4000000 SOW1	STC07224	00000090	\$HASP395 BPXAS ENDED	
19221 17:20:23.47	N 0000000 SOW1			IEA989I SLIP TRAP ID=X33	
19221 17:20:23.47	N 0000000 SOW1			\$HASP250 BPXAS PURGED --	
19221 17:20:23.73	N 0200000 SOW1	STC07223	00000090	\$HASP250 BPXAS PURGED --	
19221 17:20:23.73	N 0200000 SOW1	STC07224	00000090	\$HASP100 BPXAS PURGED --	
19221 17:28:00.87	N 0200000 SOW1	STC07242	00000281	\$HASP373 BPXAS ON STC	
19221 17:28:01.22	N 4000000 SOW1	STC07242	00000090	\$HASP100 BPXAS STARTED	
19221 17:28:01.23	N 4000000 SOW1	STC07242	00000090	BPXP0241 BPXAS INITIATOR 0060	
19221 17:28:03.56	N 0200000 SOW1	STC07243	00000281	\$HASP100 BPXAS ON STC	
19221 17:28:03.89	N 4000000 SOW1	STC07243	00000090	\$HASP373 BPXAS STARTED	
19221 17:28:03.89	N 0000000 SOW1	STC07243	00000090	BPXP0241 BPXAS INITIATOR 0070	
19221 17:28:47.71	N C0000000 SOW1	INTERNAL	00000290	S SMFDUMPS, MAN= 'SYS1.SOW'	
19221 17:28:47.71	N C0200000 SOW1			SMF SWITCH CAUSED COMMAN	
19221 17:28:47.71	N 0000000 SOW1			N 0000000 SOW1 19221 17:28:47.71 00000290 IEF196I SMF SWITCH CAUSE	
19221 17:28:47.71	N 4000000 SOW1			IEE388I SMF NOW RECORDING	
19221 17:28:47.71	N 8000000 SOW1		724 00000090	17.28.47	
19221 17:28:47.80	N 0200000 SOW1			*IEE985A SMF IS PROCESSING	
19221 17:28:48.16	N 0020000 SOW1	STC07244	00000281	\$HASP100 SMFDUMPS ON STC	
19221 17:28:48.16	N 4000000 SOW1	STC07244	00000090	\$HASP373 SMFDUMPS STARTED	
19221 17:28:48.16	N 4000000 SOW1	STC07244	00000090	\$HASP395 SMFDUMPS ENDED	
19221 17:28:50.38	N 0000000 SOW1			IEA989I SLIP TRAP ID=X33	
19221 17:28:50.38	N 0000000 SOW1			00000290	
***** BOTTOM OF DATA *****					
F1=HELP	F2=SPLIT	F3=END	F4=RETURN	F5=IFIND	F6=BOOK
F7=UP	F8=DOWN	F9=SWAP	F10=LEFT	F11=RIGHT	F12=RETRIEVE

Feel free to take some time investigating the logs. It may take a few minutes before some of it makes sense.

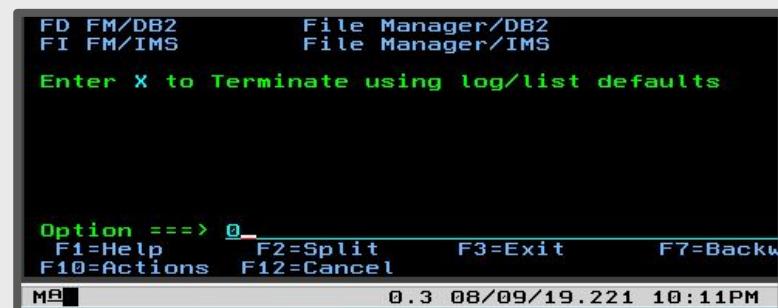
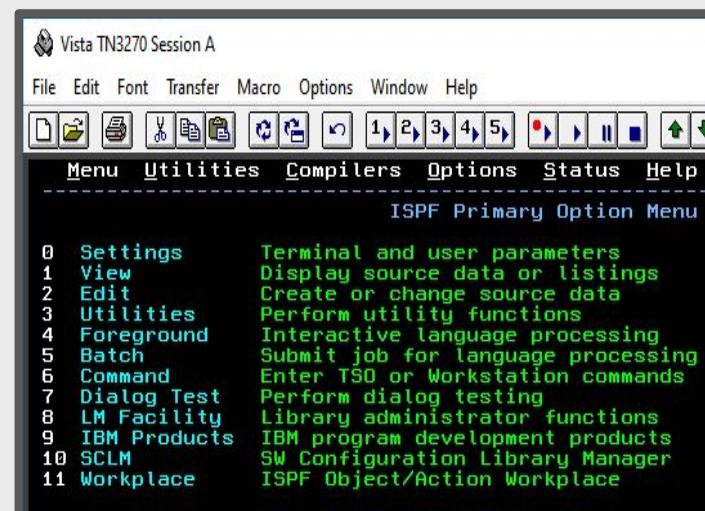
When you are ready, navigate back to the main ISPF screen by hitting **F3** twice.

Customizing Your Environment

Moving the Command Line

The primary options menu indicates everything available within this ISPF for you. Option 0 is settings.

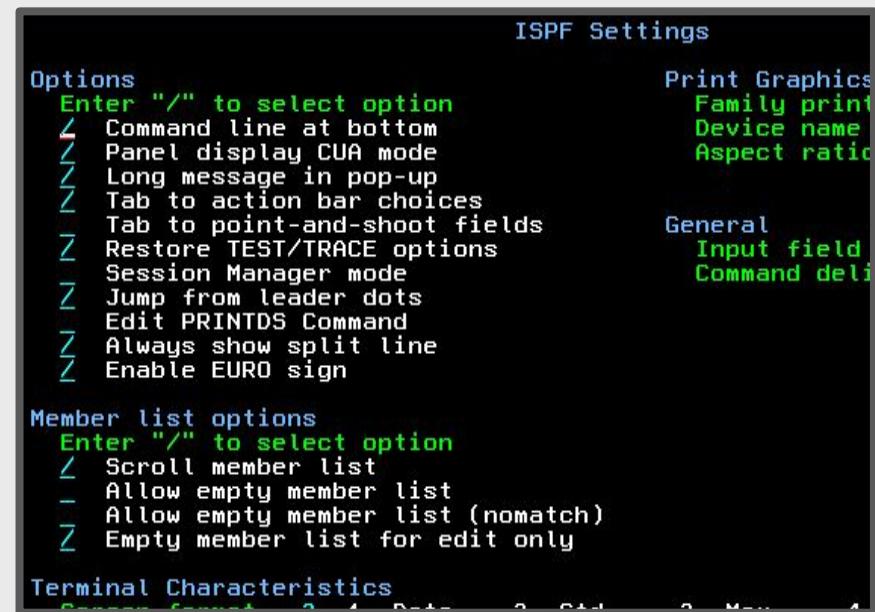
1. Type **0** to select the Settings menu in the command prompt at the bottom.
2. Press **ENTER** to access to Settings ISPF.



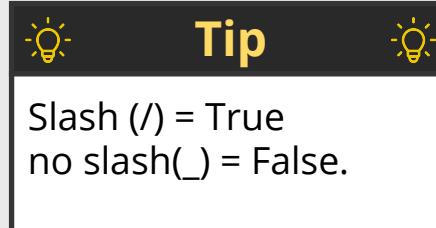
Customizing Your Environment

Moving the Command Line

1. Navigate to the option *Command line at bottom*.
2. Typing in the `/` symbol in the empty space (shown in screenshot) acts like a **True or False** statement.
3. Uncheck this setting by typing **SPACE** over the `/` symbol.



The screenshot shows the ISPF Settings menu. The 'Options' section contains several checkboxes, one of which is checked and highlighted with a red underline. The option is labeled 'Command line at bottom'. To the right of the menu, there are sections for 'Print Graphics', 'General', and 'Input field' with various sub-options listed.



Customizing Your Environment

Moving the Command Line

4. Once unchecked, navigate back using the **F3** button.



Customizing Your Environment

Resizing the Console

You may want to resize your 3270 Terminal to fit more characters on the screen or to increase/decrease the font size.

You may have also tried to drag your screen larger and noticed that it didn't behave the way you expected it to. To change the dimensions and appearance of your terminal, we will need to head into the Screen Format option of your 3270 emulator.

Customizing Your Environment

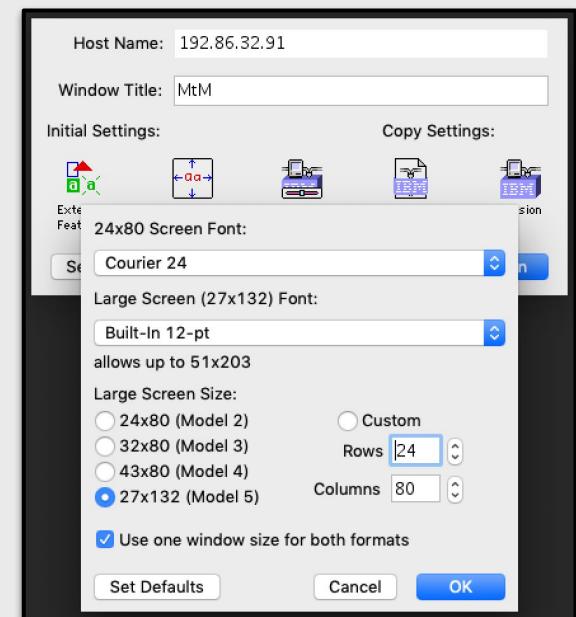
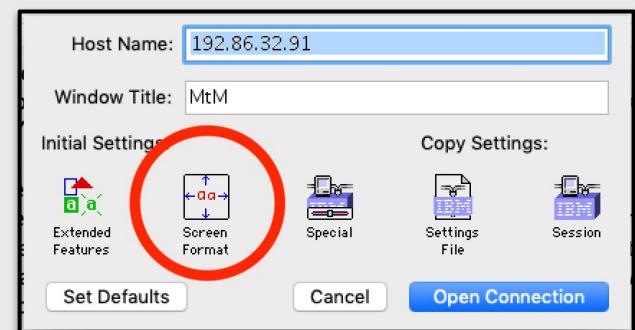
Resizing the Console

Screen Size: When you are first connecting to the system, you can set up the Screen Format.

There are different screen “modes”, which determine the number of characters that fit on the screen at once.

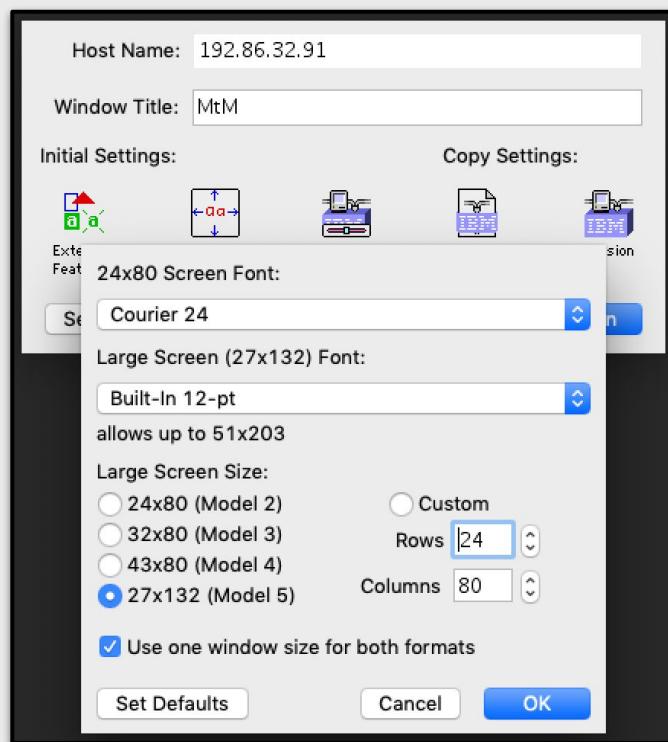
The default, **Mode 1**, can fit 24 rows (down) and 80 columns (across), while **Mode 5** supports 27 rows and 132 columns.

(Note: The screenshots are from TN3270 on Mac. For Vista 3270, these options appear on the connection panel, and in x3270 for Linux, they are under the Option menu, though the wording is slightly different)



Customizing Your Environment

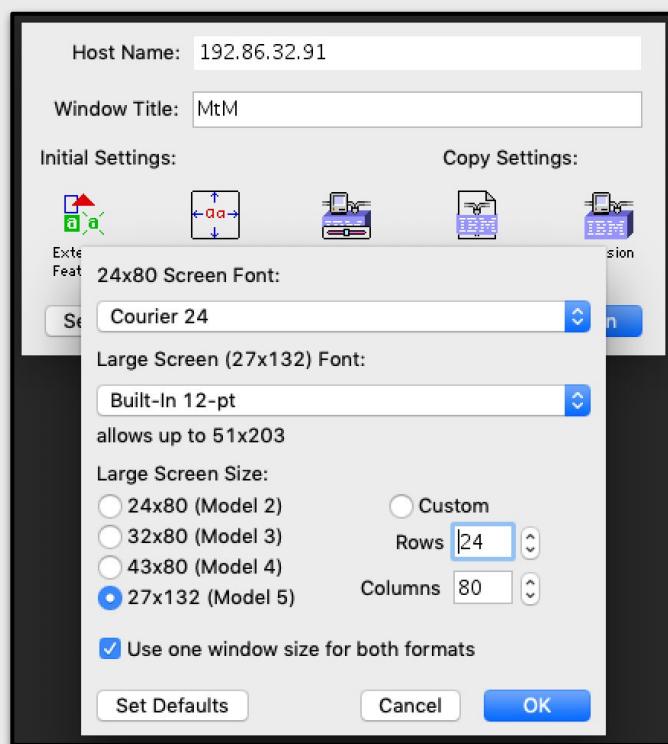
Resizing the Console



Setting a larger number of rows and/or columns will allow you to fit more text on the screen, which means less scrolling and paging around, but may make certain panels harder to read, so you may want to spend some time experimenting, but changing screen sizes typically requires re-connecting to the system, so be sure you're ready to disconnect and log back in before making changes.

Customizing Your Environment

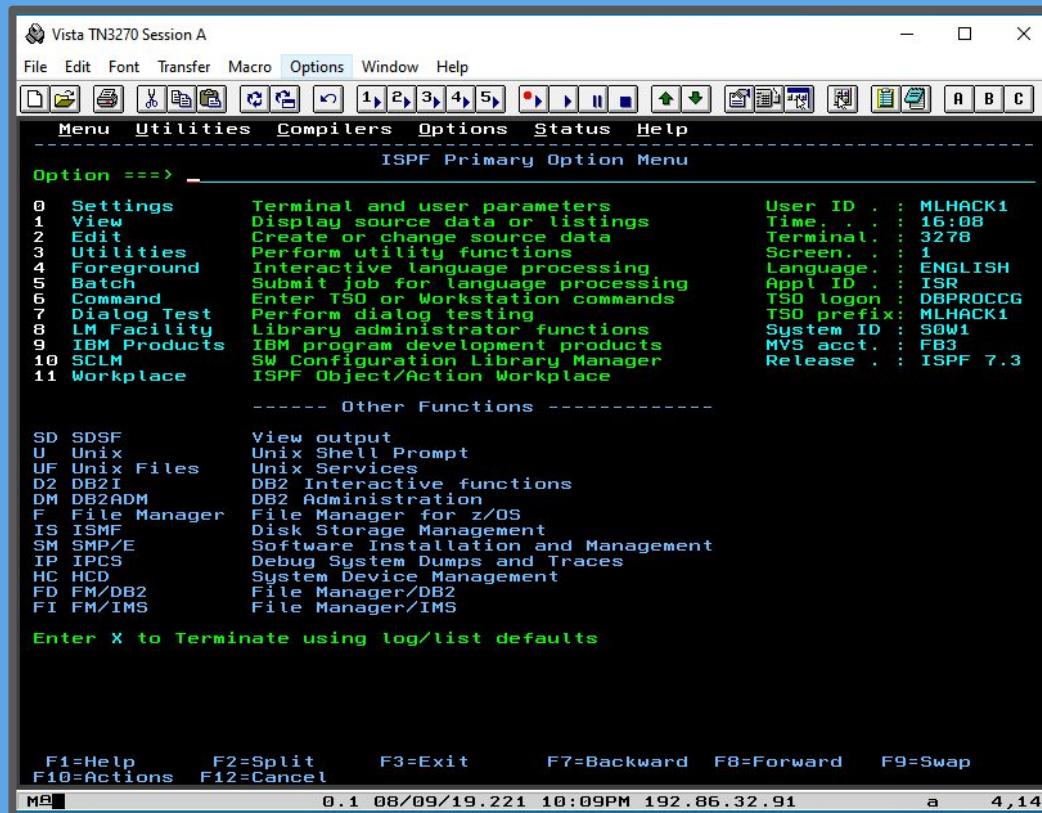
Changing the Font



Font: You can change the typeface and size used for the terminal. For optimum readability, try to stick to standard monospaced fonts such as Courier, Lucidia Console, Menlo, Monaco, or pretty much anything with “Mono” in the name or description. You should be able to switch font size without restarting your connection.

Hacking Customizations

Check it out! Your command line is now at the top, your screen is larger, and you may have changed your font!



Logging off TSO

When you want to exit your Mainframe time sharing option, you will need to log out.

You **do not** need to still be connected via your 3279 terminal for a TSO to remain active.

A TSO will stay up and running for several minutes if you **haven't** explicitly logged off.

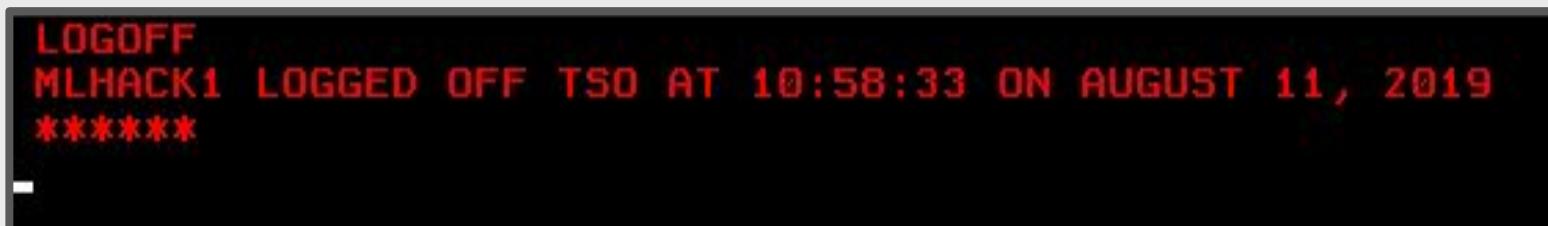
After approximately 3 minutes, it will deactivate automatically and free up Mainframe resources.

Logging off TSO

Logging off is quick and simple.

Let's do it from the main ISPF screen inside your TSO.

1. Make sure you navigate back to your TSO ISPF.
2. Press **F3** on your keyboard.
3. You will log out, and receive a message telling you so.



```
LOGOFF
MLHACK1 LOGGED OFF TSO AT 10:58:33 ON AUGUST 11, 2019
*****
```

What happens if I forget?

If you don't remember to manually log off

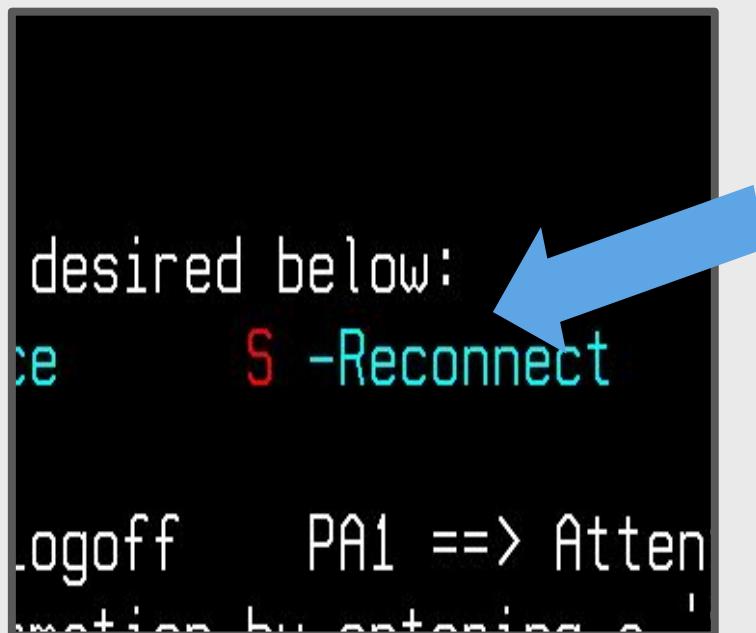
- Your TSO environment will eventually be deallocated and your account logged off.
- This will happen after 3 minutes of being inactive in your TSO. This helps you avoid using up unintentional Mainframe resources.

Key Term

Deallocated: to remove from a set of resources. In the mainframe, when logging out the dedicated environment that was given to you is removed.

Using TSO -Reconnect

If you forget to log off before closing the emulator *after* your TSO environment has been allocated, you may need to select the **-Reconnect** option on the TSO logon screen when attempting to get back into TSO.



Remember this setting we saw previously?

What is a checkpoint?

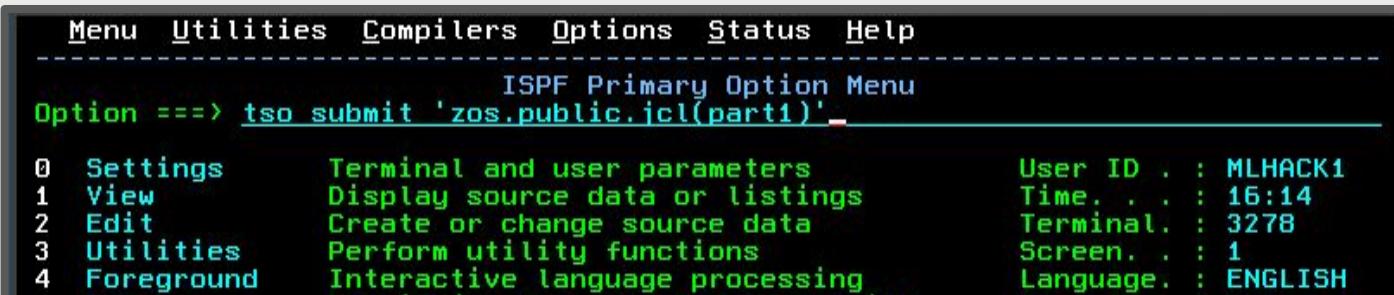
Checkpoints are logical restart points in a job which help in restarting a step from a point instead of restarting from the beginning.

When you set up checkpoints for individual steps, the status of the executing program is recorded periodically in a data set assigned for that purpose.

You can think of them similarly to checkpoints in a video game, they are like micro saves at certain points to save you the effort of having to do everything again.

Submitting The First Checkpoint

1. Log back in to **TSO** and start up ISPF.
2. Navigate to the command line and enter:
tso submit 'zos.public.jcl(part1)'
3. What this will do is trigger a job (program) to run, that will prepare some Datasets for the next stage!



Menu Utilities Compilers Options Status Help

ISPF Primary Option Menu
Option ==> tso submit 'zos.public.jcl(part1)'

0 Settings Terminal and user parameters User ID . : MLHACK1
1 View Display source data or listings Time. . . : 16:14
2 Edit Create or change source data Terminal. . : 3278
3 Utilities Perform utility functions Screen. . : 1
4 Foreground Interactive language processing Language. . : ENGLISH

You submitted your first checkpoint!

You will be prompted to enter jobname character(s) - enter: [a](#)

```
ISPF system data set allocation error - press Enter to continue.  
Log file allocation error - ISPF will operate without a log data set.  
Already cataloged, VSAM protected, or other - 'MLHACK1.S0W1.SPFL0G1.LIST'.  
ENTER JOBNM CHARACTER(S) -
```

a_-

And you'll get an acknowledgement of your submittal. Nice work!

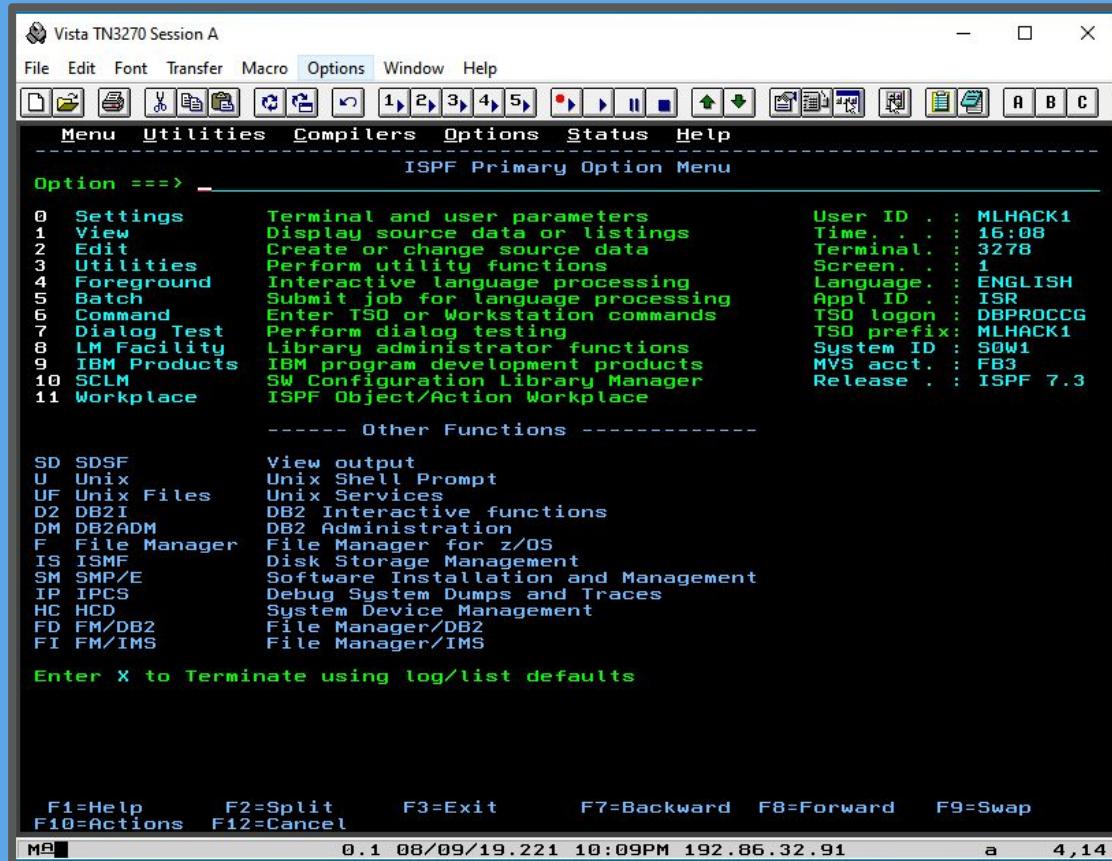
```
ISPF system data set allocation error - press Enter to continue.  
Log file allocation error - ISPF will operate without a log data set.  
Already cataloged, VSAM protected, or other - 'MLHACK1.S0W1.SPFL0G1.LIST'.  
ENTER JOBNM CHARACTER(S) -
```

a

```
JOB MLHACK1A(JOB07234) SUBMITTED  
*** -
```

Now let's work with data

You covered some basic navigation and user input and environment modding. Now let's create something!



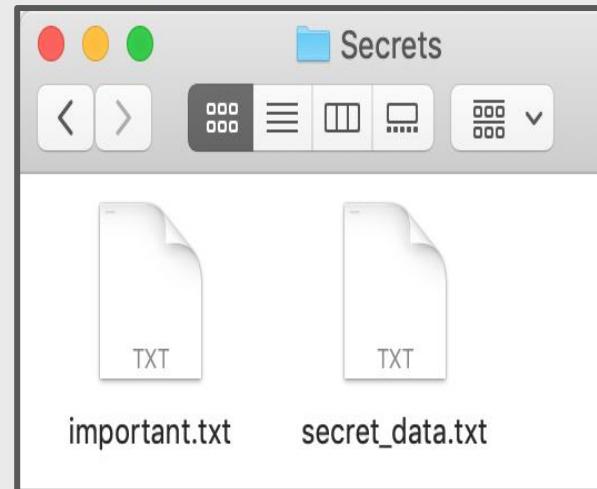
Storing Data on a Mainframe

If you use a computer regularly, you are already familiar with how files work!

Files contain data



Folders can contain files



Data Sets

A file in z/OS is called a **Data Set**.

There are two types of data sets we will encounter today, and they have a slightly different functions from each other. But they are both used for collecting or storing information of some type.

We have:

- **Sequential** Data Set
- **Partitioned** Data Sets

Sequential Data Set

- Stored items of data that can consist of anything! Much like cooking instructions, or a simple text file with your favorite movies inside.
- The data must be parsed *sequentially*. This means if 20 items exist in a sequential data set, and you want item 11. You must pass the preceding 10 items to access it first.
- Records inside data sets typically match in length. This level of definition makes for very efficient and high performing dataset access.

Partitioned Data Set

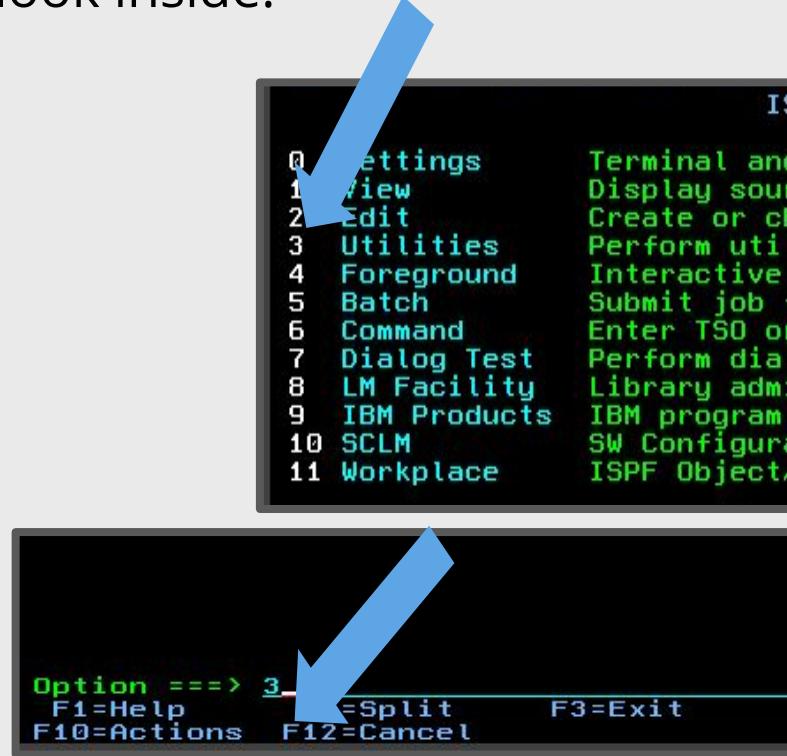
- These data sets are more representative of **files** inside of a **folder**.
- Keeps an **index** (detailed record) of all the members inside, and let's users reference and access them by their reference name.
- The members inside a partitioned dataset are often **sequential data sets**.

Browsing Data Sets

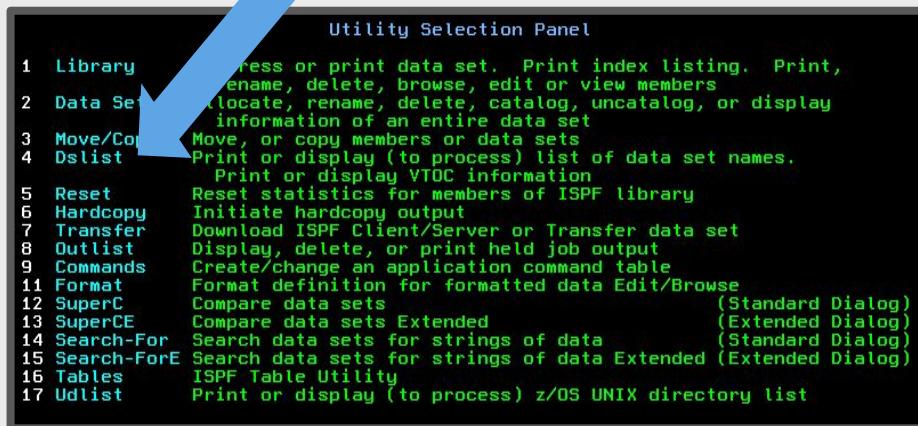
ISPF provides a facility to browse and inspect data sets on the z/OS filesystem.

Let's navigate the ISPF and take a look inside!

1. Log back on to ISPF.
2. In the primary options, identify the option **Utilities**. It is item 3 in the command menu. Type '**3**' in the input field and press **ENTER**.



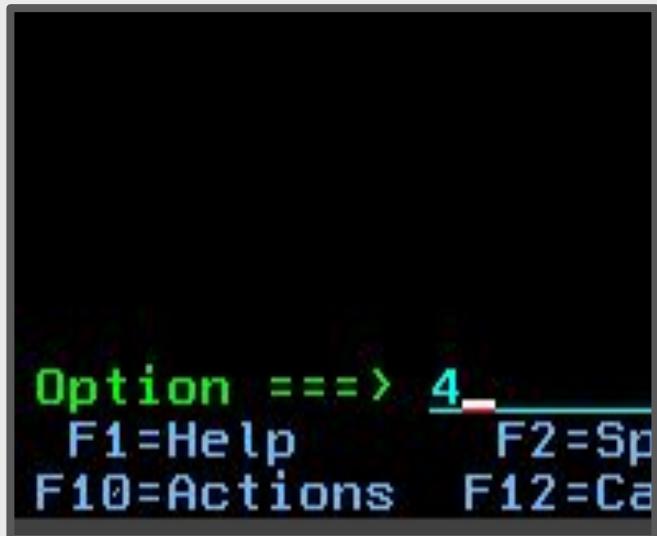
Browsing Data Sets



Utility Selection Panel

1 Library	Access or print data set. Print index listing. Print, rename, delete, browse, edit or view members
2 Data Set	Locate, rename, delete, catalog, uncatalog, or display information of an entire data set
3 Move/Copy	Move, or copy members or data sets
4 Dslist	Print or display (to process) list of data set names. Print or display VTOC information
5 Reset	Reset statistics for members of ISPF library
6 Hardcopy	Initiate hardcopy output
7 Transfer	Download ISPF Client/Server or Transfer data set
8 Outlist	Display, delete, or print held job output
9 Commands	Create/change an application command table
11 Format	Format definition for formatted data Edit/Browse
12 SuperC	Compare data sets (Standard Dialog)
13 SuperCE	Compare data sets Extended (Extended Dialog)
14 Search-For	Search data sets for strings of data (Standard Dialog)
15 Search-ForE	Search data sets for strings of data Extended (Extended Dialog)
16 Tables	ISPF Table Utility
17 Uplist	Print or display (to process) z/OS UNIX directory list

You will be presented with another similar menu of utilities. Here you can access several tools for **filtering** and **searching datasets by a term**.

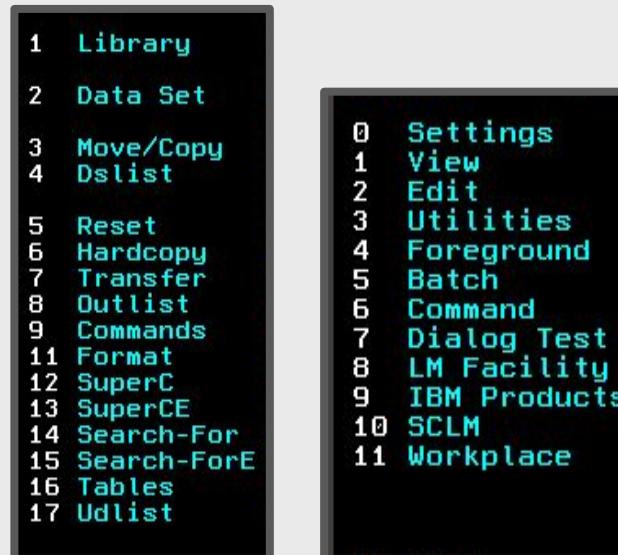


1. The one we are interested now is **Dslist**, which stands for **Data Set List**. You will notice it is option 4.
2. Enter **4** and press **ENTER**.

z/OS Utilities

Instead of using a mouse and cursor to move from utility to utility, we use numbers and letters to navigate around. This may seem cumbersome at first, but with familiarity, you will be able to navigate around very quickly.

The **utilities** menus provide an amazing way of performing many simple and complex actions.



The image shows two separate terminal windows displaying menu lists. The left window has a light gray background and lists numbered options from 1 to 17. The right window has a black background and lists numbered options from 0 to 11. Both windows use blue text for the menu items.

Left Window (Light Gray)	Right Window (Black)
1 Library	0 Settings
2 Data Set	1 View
3 Move/Copy	2 Edit
4 Dslist	3 Utilities
5 Reset	4 Foreground
6 Hardcopy	5 Batch
7 Transfer	6 Command
8 Outlist	7 Dialog Test
9 Commands	8 LM Facility
11 Format	9 IBM Products
12 SuperC	10 SCLM
13 SuperCE	11 Workplace
14 Search-For	
15 Search-ForE	
16 Tables	
17 Udlist	

Dataset naming convention

- On z/OS, the names of datasets are comprised of several small strings, separated by a '.' character.
- Each of these small strings can **not** be longer than 8 characters.
- In this environment, you will generally see your username as the first string in the sequence of strings that make up a dataset name.

Dataset naming convention

Example:

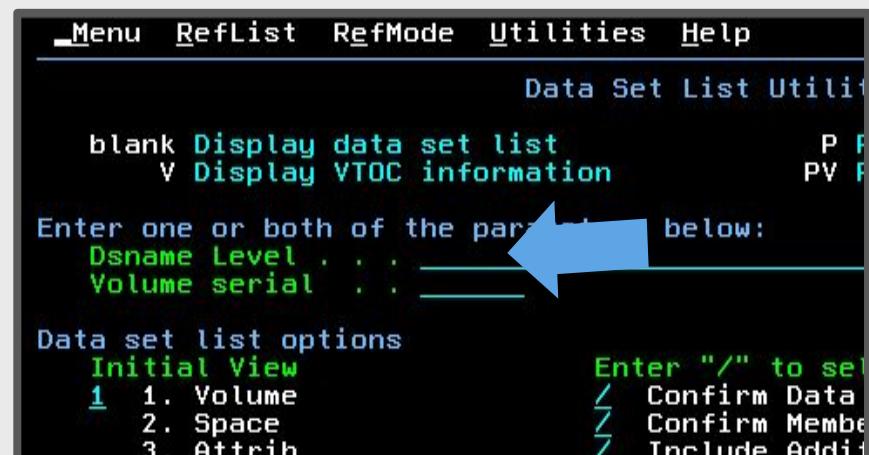
If your username was **MLHACK**.

You would find your datasets to be named such as

MLHACK . SEQ . DATA.

This would represent the user **MLHHACK** has access.
The dataset they have access to is a **sequential dataset**.

Browsing Data Sets



Let's investigate any data sets that are associated with your user ID.

1. Start by locating the field titled **Dsname Level**.
2. Enter your User ID in the empty **Dsname** field.



Reminder, your User ID is the one you logged into the Mainframe with!

Browsing Files

At the top of the interface the **Data Set List Utility** indicates that we should *leave* the command input **blank** to show a list of data sets.

- Once your userID is typed in Dsname field, press **ENTER** to proceed.



```
Menu RefList RefMode Utilit
Data
blank Display data set list
  V Display VTOC information

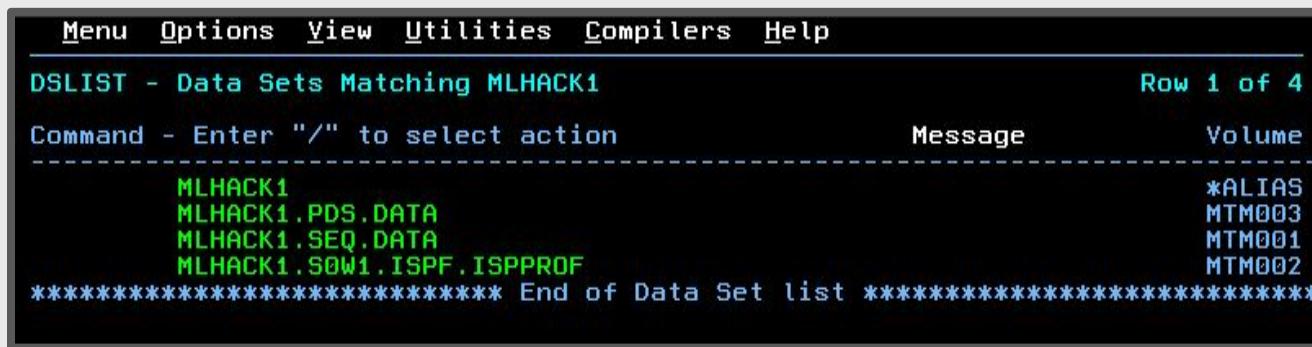
Enter one or both of the parameters
  Dsname Level . . . MLHACK1_
  Volume serial . . .

Data set list options
  Initial View
  1 1. Volume
  2. Space
  3. Attrib
  4. Total

When the data set list is displayed
  "/" on the data set list command
  an ISPF line command, the name
  "=" to execute the previous co
```

Browsing Data Sets

As the ISPF changes you will see a list of data sets.



The screenshot shows an ISPF menu with the following options: Menu, Options, View, Utilities, Compilers, Help. Below the menu is a title 'DSLIST - Data Sets Matching MLHACK1' and a message 'Row 1 of 4'. A command line says 'Command - Enter "/" to select action'. The main area displays a list of data sets with columns: Message, Volume, and a list of datasets. The datasets listed are: MLHACK1, *ALIAS; MLHACK1.PDS.DATA, MTM003; MLHACK1.SEQ.DATA, MTM001; and MLHACK1.SOW1.ISPPROF, MTM002. At the bottom, a footer message reads '***** End of Data Set list *****'.

Message	Volume
MLHACK1	*ALIAS
MLHACK1.PDS.DATA	MTM003
MLHACK1.SEQ.DATA	MTM001
MLHACK1.SOW1.ISPPROF	MTM002

The list can be recognised as the green text with items prefixed with **MLHACK1**. Yours will look slightly different!

The first item in the list is just our account **ALIAS**. But let's look below!

Browsing Data Sets

We have two readily accessible datasets!

Let's think back to the dataset naming conventions mentioned before.

```
-----  
MLHACK1  
MLHACK1.PDS.DATA  
MLHACK1.SEQ.DATA  
MLHACK1.SOW1.ISPF.ISPPROF  
***** End of Data
```

MLHACK1.PDS.DATA - this name suggests that it is a **Partitioned Data Set**.

MLHACK1.SEQ.DATA - this name suggests that it is a **Sequential Data Set**.

Sequential Data Sets

```
Menu Options View Utilities Comp  
DSLIST - Data Sets Matching MLHACK1  
Command - Enter "/" to select action  
-----  
MLHACK1  
MLHACK1.PDS.DATA  
MLHACK1.SEQ.DATA  
MLHACK1.S0W1.ISPF.ISPPROF  
***** End of Data
```



We can open and look inside these datasets inside our the ISPF we're using now.

1. Navigate to the input field on the left side of the **USERID.SEQ.DATA** dataset.
2. Type in **b** and press **ENTER**.

```
Command - Enter "/" to select action  
-----  
MLHACK1  
MLHACK1.PDS.DATA  
b- MLHACK1.SEQ.DATA  
MLHACK1.S0W1.ISPF.ISPPROF  
***** End of Data
```

The '**b**' command represents a '**browse**' function.

It's the Simpsons!

Sequential Data Sets

- This is the inside of a **Sequential Data Set**. It is the closest representation of a conventional 'file' on z/OS. It is just a block of data, with a predefined length.
- Every **Sequential Data Set** is made of **RECORDS** of predefined length. Think of **RECORDS** as the lines of text. In this dataset, each line is arbitrary symbols that build an image of the Simpsons characters. But it can be anything, from medical to banking data!

The screenshot shows a mainframe terminal window with a dark background. It displays several lines of data representing records in a sequential dataset. The data is formatted as ASCII art of the Simpson family: Homer (a blocky character), Marge (a woman with a large head), Bart (a young boy with a large head), and other characters like Moe and Krusty. The data is preceded by a header with file and record information. At the bottom of the screen, there is a command prompt 'Command ==>' followed by a series of function key definitions:

F1=Help	F2=Split	F3=Exit	F5=Rfind	F7=Up
F10=Left	F11=Right	F12=Cancel		

Simpsons ASCII

This Simpsons artwork inside a dataset is comprised of characters found entirely on a computer, just like the ones you're using to control the ISPF and utilities.

This art style is called **ASCII-Art**.

Sequential Data Sets

Inside this dataset, you can navigate using **F7** and **F8** to move Up and Down the document, just like we did with the **LOGS** earlier in the workshop.

The **Scroll** field indicates how many records the 'scroll' function moves up or down at once. You can change it from its default, which is **PAGE** to a number.

This number will show how many individual lines are moved up or down every time you press **F7** or **F8** to navigate.



Control Keys



When navigating inside datasets, remember that the control panel is listed at the bottom of the ISPF.

After you have been inside a dataset, you will probably want to **Exit** the file eventually.

You can press **F3** for this.

Partitioned Data Sets

Partitioned Data Sets are capable of containing many members, which can be treated like **Sequential sub-data sets**.



```
Menu Options View Utilities Comp  
DSLIST - Data Sets Matching MLHACK1  
Command - Enter "/" to select action  
-----  
MLHACK1  
MLHACK1.PDS.DATA  
MLHACK1.SEQ.DATA  
MLHACK1.S01 ISPF.ISPPROF  
***** End of D
```

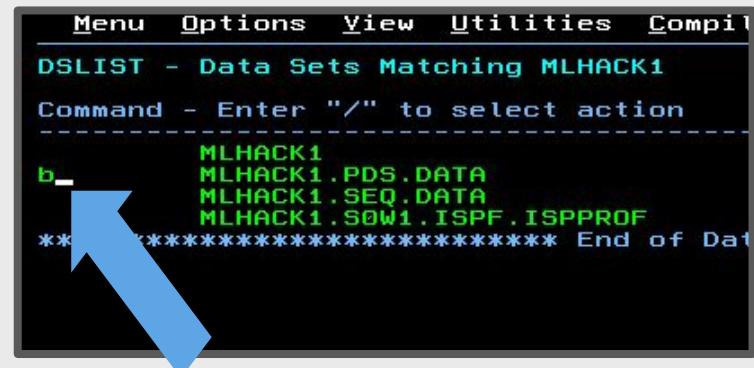
This might remind you of a folder from an operating system like Windows or Mac.

A key difference between **Partitioned Data Sets** and **folders** is that they cannot also contain another **Partitioned Data Set**. A **folder** can contain as many folders as it wants!

Partitioned Data Sets

Let's browse the **partitioned data set** too!

1. Navigate to the input field for **USERID.PDS.DATA**.
2. Type **b** in the field and press **ENTER**.



The screenshot shows a terminal window with a menu bar at the top. The menu bar includes 'Menu', 'Options', 'View', 'Utilities', and 'Compil'. Below the menu is a title 'DSLIST - Data Sets Matching MLHACK1'. A command line says 'Command - Enter "/" to select action'. A list of datasets is displayed, starting with 'MLHACK1' and including 'MLHACK1.PDS.DATA', 'MLHACK1.SEQ.DATA', and 'MLHACK1.S0W1.ISPF.ISPPROF'. At the bottom, a message reads '** **** End of Data ***'. A large blue arrow points from the text 'Type b in the field and press ENTER.' in the slide content to the letter 'b' in the terminal's input field.

```
Menu Options View Utilities Compil
DSLIST - Data Sets Matching MLHACK1
Command - Enter "/" to select action
-----
MLHACK1
MLHACK1.PDS.DATA
MLHACK1.SEQ.DATA
MLHACK1.S0W1.ISPF.ISPPROF
** **** End of Data ***
```

Partitioned Data Sets

MLHACK1.PDS.DATA					
Name	Prompt	Size	Created	Changed	ID
BINARY		28	2019/05/25	2019/05/25 17:24:52	IBMUSER
COUNTRY					
FUN		21	2019/05/24	2019/05/25 14:48:51	IBMUSER
Z		23	2019/05/24	2019/05/24 16:52:46	IBMUSER
End					

Inside the the sequential data set, we were presented with clear direct text contents. Inside a partitioned dataset it is **sub-structure** of **members**.

- Inside this partitioned dataset, every member inside can be browsed just like a sequential dataset.
- You can navigate in front of these members, type **b** and press **ENTER** to also navigate inside and view the contents.

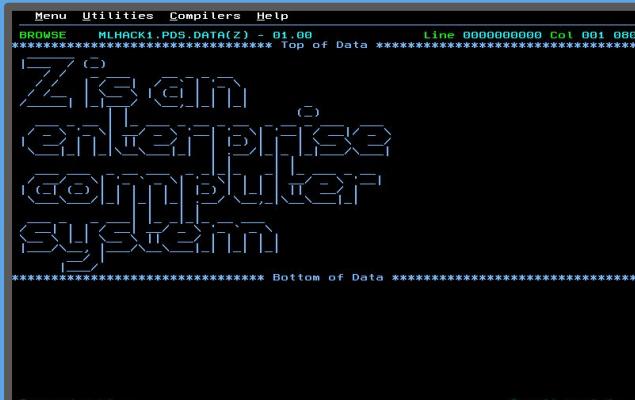
Check them out!

Take a moment to navigate around
and just exit with

F3

when you're done!

So many data sets!



```
Menu Utilities Computers Help
BROWSE MLHACK1.PDS.DAT(BINARY) - 01.00 Line 0000000000 Col 001 000
***** Top of Data *****

All data is stored in binary strings of 0's and 1's at the lowest level. However, humans do not enter 1's and 0's to enter data or write programs.

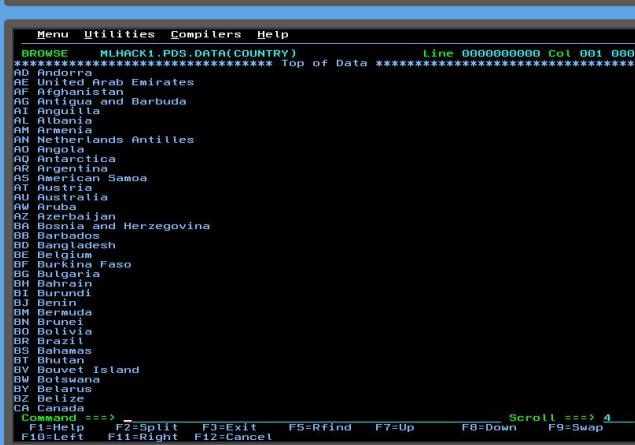
The lowest level is known as binary.
Each binary position is a bit.
A bit is 1 (on) or 0 (off).

Humans enter characters translated to a string of bits.
Computers translate characters to a string of bits and vice versa.

Hexadecimal numerals are used by computers to provide a more
human-friendly representation of binary-coded values.

Hexadecimal numerals are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
and 10, 11, 12, 13, 14, 15 where
-- 0 is decimal 0
-- B is decimal 11
-- D is decimal 13
-- E is decimal 14
-- F is decimal 15

Binary 0000 is hexadecimal 0 and decimal 0
Binary 1111 is hexadecimal F and decimal 15
***** Bottom of Data *****
```



The screenshot shows a menu bar with 'Menu Utilities Compilers Help'. Below it, the title 'BROWSE MHACK1.PDS.DAT(FUN) - 01.02' is displayed, along with 'Line 0000000000 Col 001 000'. The main area contains several rotated and mirrored versions of the word 'EFTI WALL' in a blocky, pixelated font. Below these, a question is posed: 'Do you know the Efti Wall word below?'. At the bottom, the text 'Bottom of Data' is preceded by a series of rotated and mirrored characters.

Hello, Unix Filesystem

So far, we've been using ISPF to interact with z/OS, but the z/OS Operating System also provides a UNIX interface so you can run UNIX commands and work within a UNIX filesystem.

Even better, you can manage z/OS datasets from within the UNIX environment and vice-versa.

In the next challenge, we are going to create a file on the **unix filesystem**, and manipulate it on the z/OS filesystem.

What's a Unix File system?

- It is an alternate file system to z/OS
- It's a central component in the software of many desktop and server environments.
- It provides a specific format for organising **files** and **folders** into a tree structure hierarchy.
- It is very recognisable file structure in Linux and Mac computers.

For example

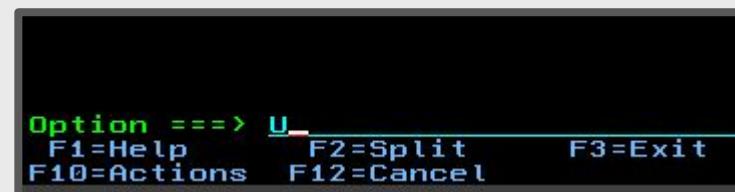
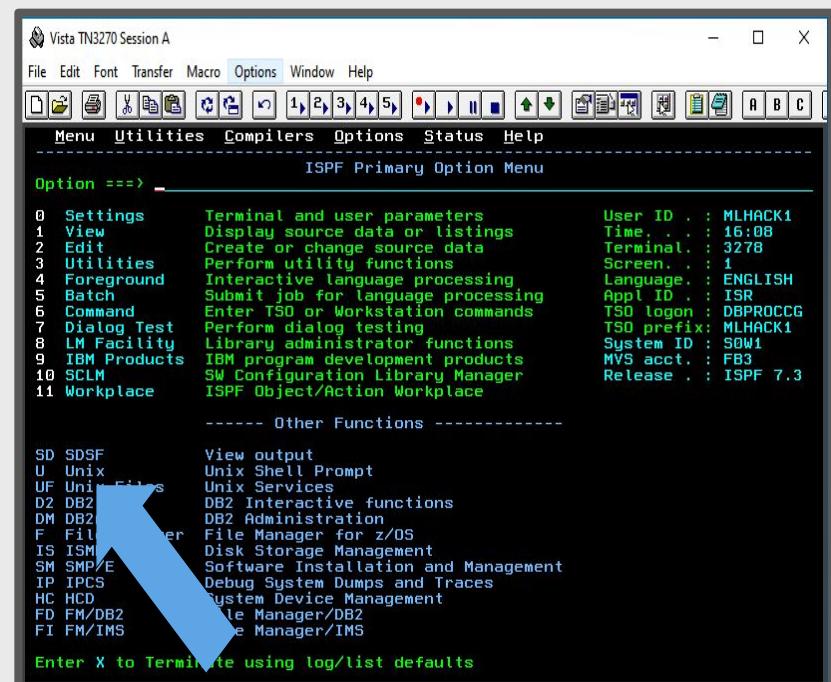
In a Unix environment:

*Personal files and folders would go into a folder called **/home**.
Executable binaries would go into one folder called **/bin**.*

Back to ISPF Main Menu

Let's navigate back in the ISPF Main Menu. Here you will find an option named **Unix (Unix Shell Prompt)** in Other Functions.

1. You can activate this by typing **U** and pressing **ENTER**.



Unix Shell and Filesystem

This interface is different from the ISPF one you've been using

This is a **Unix Shell** interface

You're going to use this to explore the *Interoperability* of z/OS and UNIX.

```
IBM
Licensed Material - Property of IBM
5650-ZOS Copyright IBM Corp. 1993, 2017
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo.

U.S. Government Users Restricted Rights -
Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

/z/mlhack1 >
```

Note: You may need to press *ENTER* again as a response to a *** prompt in the input field.

Unix Shell and Filesystem

When you activate UNIX, your terminal interface will open itself into a location of a UNIX Filesystem. The directory you are placed inside is in the format of **/z/userid**.

This is your home on the unix system.



```
IBM
Licensed Material - Property of IBM
5650-ZOS Copyright IBM Corp. 1993, 2017
(C) Copyright Mortice Kern Systems, Inc., 1985, 199
(C) Copyright Software Development Group, Universit

U.S. Government Users Restricted Rights -
Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

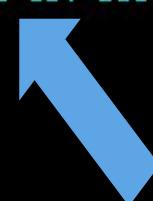
/z/mlhack1 >
```

Unix Shell and Filesystem

To get started with the Unix file system, let's create a standard unix file next..

1. Make sure your cursor is in the input field.
2. Type in **date** and press ENTER.

This is going to display the **current date** and **time** *live* in your Terminal!



```
IBM
Licensed Material - Property of IBM
5650-ZOS Copyright IBM Corp. 1993, 2017
(C) Copyright Mortice Kern Systems, Inc., 198
(C) Copyright Software Development Group, Uni

U.S. Government Users Restricted Rights -
Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp

/z/mlhack1 > date
Wed Aug 14 19:15:45 CDT 2019
/z/mlhack1 >
```

Unix Shell and Filesystem

Often with files and computer processes we save the time and date somewhere to know when something happened!

We might store this in the file name to know *when it was created*, or inside to tell us *when the file was modified*. This is typically known as **Timestamping**.

What we will do next is save a timestamp, inside a file on the Unix system.



Unix Shell and Filesystem

We need to create a file and fill a file with date and time as its contents.

1. Navigate to the input field.
Type in **date > p1** and press **ENTER**.

This is going to print the **datetime** but will redirect it from the Terminal output to a file named **p1**. Since this file doesn't yet exist, Unix will create it for you!

```
Use, duplication or disclosure  
GSA ADP Schedule Contract wit  
  
IBM is a registered trademark  
  
/z/mlhack1 > date  
Wed Aug 14 19:15:45 CDT 2019  
/z/mlhack1 > date > p1  
/z/mlhack1 >
```

Unix Shell and Filesystem

A Unix command called **Cat** exists to read the contents of a file and print it in the Terminal. We can use this to check our file was created and the contents are inside!

1. Type in **cat p1**.



```
z/m1hack1 > date
ed Aug 14 19:15:45 CDT 2019
z/m1hack1 > date > p1
z/m1hack1 > cat p1
ed Aug 14 19:18:13 CDT 2019
z/m1hack1 >
```

The name **cat** derives from **concatenate**. Which is to join two strings together!

Unix Shell and Filesystem

Let's get *moving!*

We have created an item in the Unix file system, it needs to be rehomed in the z/OS file system.

This can be done with more Unix commands, such as [Copy/CP](#).

Unix Shell and Filesystem

You are going to copy your timestamp file into a partitioned data set on z/OS.

This way we know where it's saved on the Mainframe. Inside **USERID.PDS.DATA**.

1. Type

```
cp p1 '//pds.data(p1)'
```

This will copy p1 through the shared space of the two filesystems!

```
IBM
Licensed Material - Property of IBM
5650-ZOS Copyright IBM Corp. 1993, 2
(C) Copyright Mortice Kern Systems,
(C) Copyright Software Development G
U.S. Government Users Restricted Rig
Use,duplication or disclosure restri
GSA ADP Schedule Contract with IBM C
IBM is a registered trademark of the
/z/mlhack1 > date
Wed Aug 14 19:15:45 CDT 2019
/z/mlhack1 > date > p1
/z/mlhack1 > cat p1
Wed Aug 14 19:18:13 CDT 2019
/z/mlhack1 > cp p1 '//pds.data(p1)'
/z/mlhack1 >
```

Unix Shell and Filesystem

It's also possible to access the dataset that is now inside the z/OS file system **from** the Unix file system. Unix tools will let us do this! We can use **Cat** again but with we can provide the syntax determine the file is in the z/OS system instead.

```
...  
Licensed Material - Property of IBM  
5650-ZOS Copyright IBM Corp. 1993, 2017  
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.  
(C) Copyright Software Development Group, University of Waterloo, 1989.  
  
U.S. Government Users Restricted Rights -  
Use, duplication or disclosure restricted by  
GSA ADP Schedule Contract with IBM Corp.  
  
IBM is a registered trademark of the IBM Corp.  
  
/z/mlhack1 > date  
Wed Aug 14 19:15:45 CDT 2019  
/z/mlhack1 > date > p1  
/z/mlhack1 > cat p1  
Wed Aug 14 19:18:13 CDT 2019  
/z/mlhack1 > cp p1 '//pds.data(p1)'  
/z/mlhack1 > cat '//pds.data(p1)'  
Wed Aug 14 19:18:13 CDT 2019  
/z/mlhack1 >
```

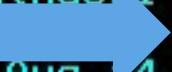
1. Type in **cat '//pds.data(p1)'**. This will display the contents of the p1 data set.

Unix Shell and Filesystem

1. Type `cat '//pds.data(p1)'`.

Notice this is slightly different from when we accessed files on z/OS and Unix.

The dataset we want to access is wrapped in single quotation marks and is prefixed with a `//`.



```
/z/mlhack1 > cat pi
Wed Aug 14 19:18:13 CDT 2019
/z/mlhack1 > cp pi '//pds.data(p1)
/z/mlhack1 > cat '//pds.data(p1)'
Wed Aug 14 19:18:13 CDT 2019
/z/mlhack1 >
```

Verify your Data Set

```
IBM
Licensed Material - Property of IBM
5650-ZOS Copyright IBM Corp. 1993, 2017
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

U.S. Government Users Restricted Rights -
Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

/z/mthack1 > date
Wed Aug 14 19:15:45 CDT 2019
/z/mthack1 > date > p1
/z/mthack1 > cat p1
Wed Aug 14 19:18:13 CDT 2019
/z/mthack1 > cp p1 '//pds.data(p1)'
/z/mthack1 > cat '//pds.data(p1)'
Wed Aug 14 19:18:13 CDT 2019
/z/mthack1 > exit

2>>> FSUM2331 The session has ended. Press <Enter> to end OMVS.
RUNNING
```

We want to check the dataset we created [P1](#) actually exists in the z/OS file system, and that we didn't do anything wrong on the way!

Let's navigate back to z/OS.

1. To leave the Unix shell, type [exit](#) and press [ENTER](#).

Verify your Data Set

```
IBM
Licensed Material - Property of IBM
5650-ZDS Copyright IBM Corp. 1993, 2017
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.

U.S. Government Users Restricted Rights -
Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

/z/mlhack1 > date
Wed Aug 14 19:15:45 CDT 2019
/z/mlhack1 > date > p1
/z/mlhack1 > cat p1
Wed Aug 14 19:18:13 CDT 2019
/z/mlhack1 > cp p1 '//pds.data(p1)'
/z/mlhack1 > cat'//pds.data(p1)'
Wed Aug 14 19:18:13 CDT 2019
/z/mlhack1 > exit

2>>> FSUM2331 The session has ended. Press <Enter> to end OMVS.
RUNNING
```

A message appears below telling you the session has ended.

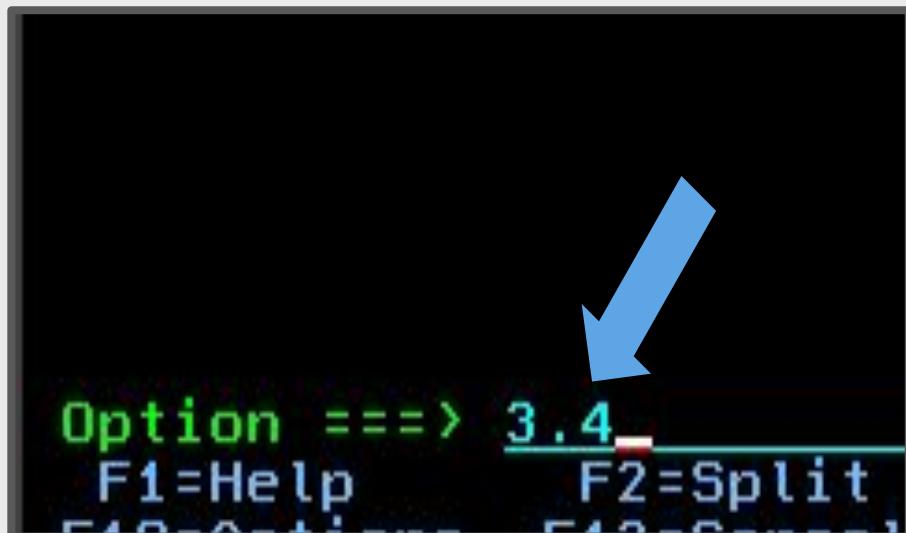
2. Press **ENTER** one more time and you will navigate back to ISPF.

Quick shortcut!

Instead of finding the Utilities menu, entering a first option, then a second, we can go directly to the **Utility** we want, which is **Dslist**!

Since we know that **Utilities** is option number **3**, and **Dslist** is option number **4** in the menu after.

Quick shortcut!



Let's try it!

3. Type **3.4** in the command input to be directly taken there.
4. Press **ENTER**.

Verify your Data Set

```
RefList RefMode Utilities Help
Data Set List Utilit
Display data set list          P P
Display VTOC information      PV P
; or both of the parameters below:
; Level . . . MLHACK1_
; serial . . .
list options
 1 View
 Volume
 Space
 Attrib
 Total
Enter "/" to set
  / Confirm Data
  / Confirm Member
  / Include Addit
  / Display Catalog
  / Display Total
  / Prefix Dsname
```



Now you're immediately taken to the **Data Set List Utility!**

1. Like before, enter your user identifier in the **Dsname field**.
2. Press **ENTER!**

Verify your Data Set

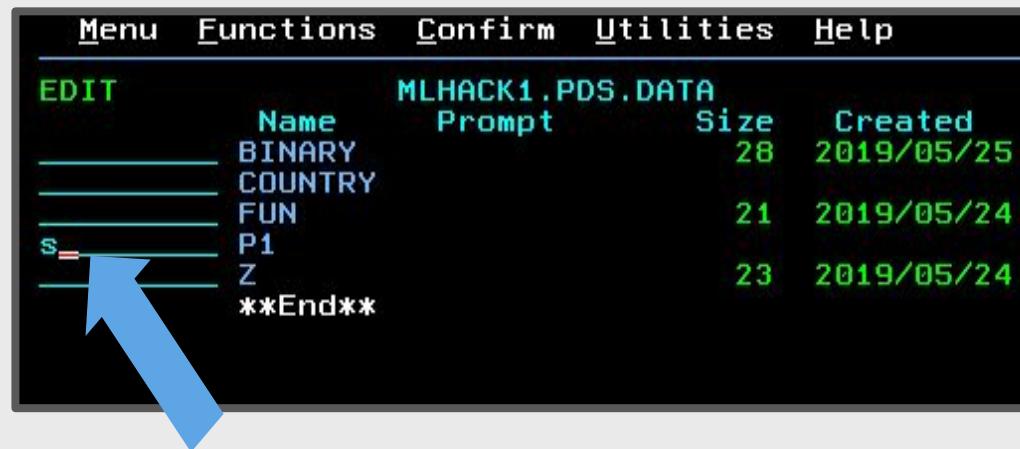
Let's use the **Edit** tool to try see the file we transferred from the Unix file system!



```
Menu Options View Utilities Com  
DSLIST - Data Sets Matching MLHACK1  
Command - Enter "/" to select action  
-----  
MLHACK1  
MLHACK1.PDS.DATA  
MLHACK1.SEQ.DATA  
MLHACK1.SOW1.ISPF.ISPPROF  
***** End of
```

1. Type in **e** (for **Edit**) in the field in front of the Partitioned Dataset **USERID.PDS.DATA**.
2. Press **ENTER!**

Verify your Data Set



This is a partitioned data set, so the utility will not know which member you want to edit. It will present you with a list of members to select.

(Psst! Your **P1** dataset should be in here!)

1. Type **s** for **Select** in front of the **P1** data set and press **ENTER**.

Verify the P1 Data Set

```
File Edit Edit_Settings Menu Utilities Compilers Test Help

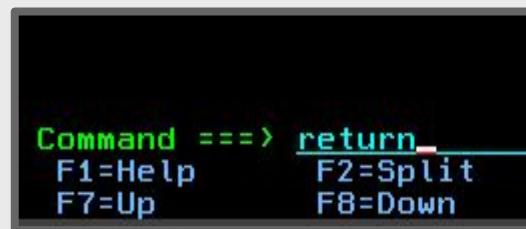
EDIT      MLHACK1.PDS.DATA(P1) - 01.00          Columns 00001 00072
***** **** Top of Data ****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 Wed Aug 14 19:18:13 CDT 2019
***** **** Bottom of Data ****
```

You should be looking at your **timestamp** in green.

This is the file you created in Unix and moved to z/OS.

You just transported a dataset between the two separate operating systems on an IBM Mainframe.

Completing this Workshop



1. Type **return** in the input field and press **ENTER**.

This will take you straight back to the ISPF Main Screen.

Completing this Workshop

We have nearly completed the hands-on portion of the workshop! However, to fully complete Part 1 and move on to the rest of the contest, we have to pass a quick quiz about the mainframe.

You'll find the answers to these questions in the full Part 1 of the contest, located at: mlhlocal.host/meet-the-mainframe

When you're ready to attempt the quiz and complete Part 1, enter **tso p1quiz** in the **ISPF** menu.



```
SD SDSF          View output
U Unix           Unix Shell Pr
Option ===> tso p1quiz
F1=Help         F2=Split      F3=
```

In the ISPF console type **tso p1quiz** and press **ENTER**.

Mainframe Success!

Congratulations!

You successfully completed Part 1

Part 2 is waiting for you

-

Wow, we've learned a lot today.

How about we put this all together?

Extra Challenges

Challenge 1

Create a file on the **unix file system** that contains ASCII-Art of your choice. Try to do this from memory this time!

You can generate your own ASCII-Art online:

mlhlocal.host/ascii-generator

Where to go from here...

1

Check your email after the workshop
For lots of resources to keep learning!

2

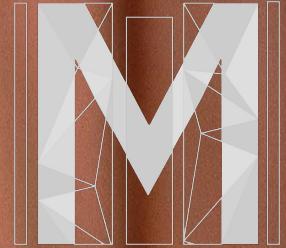
Complete the Master the Mainframe Contest
Take Part 2 and 3 of the MtM Contest and earn an official MtM Badge.

3

Sign-Up to Host your own MtM Workshop.
Host your own MtM workshop and become an recognized MtM Facilitator



The Master the Mainframe Competition has teamed up with the UN World Food Programme app, @ShareTheMeal to fight hunger!



Just 1 hour of learning feeds 2 children for a day via our team in #ShareTheMeal
masterthemainframe.com

Earn a Master the Mainframe Digital Badge



By completing Parts 2 and 3 of the Master the Mainframe Contest, you can earn the Master the Mainframe Digital Badges

Become a Recognized Master the Mainframe Facilitator



Show leadership in bringing mainframe to the next generation. By becoming a facilitator you have demonstrated your knowledge of IBM Z by sharing your experience and ability to innovate on the mainframe to solve real life situations to participants throughout workshops.

Learn more about the Earning Criteria
mlhlocal.host/become-a-facilitator

What did you learn today?

We created a fun quiz to test your knowledge and see what you learned from this workshop.

<http://mlhlocal.host/quiz>

Learning shouldn't stop when the workshop ends...

Check your email for access to:



- These workshop slides
- Practice problems to keep learning
- Deeper dives into key topics
- Instructions to complete the Master the Mainframe contest
- Instructions to become a recognized Master the Mainframe facilitator
- More opportunities from MLH!



Workshop

What the Hack is the Mainframe?

MLH localhost

IBM Z

