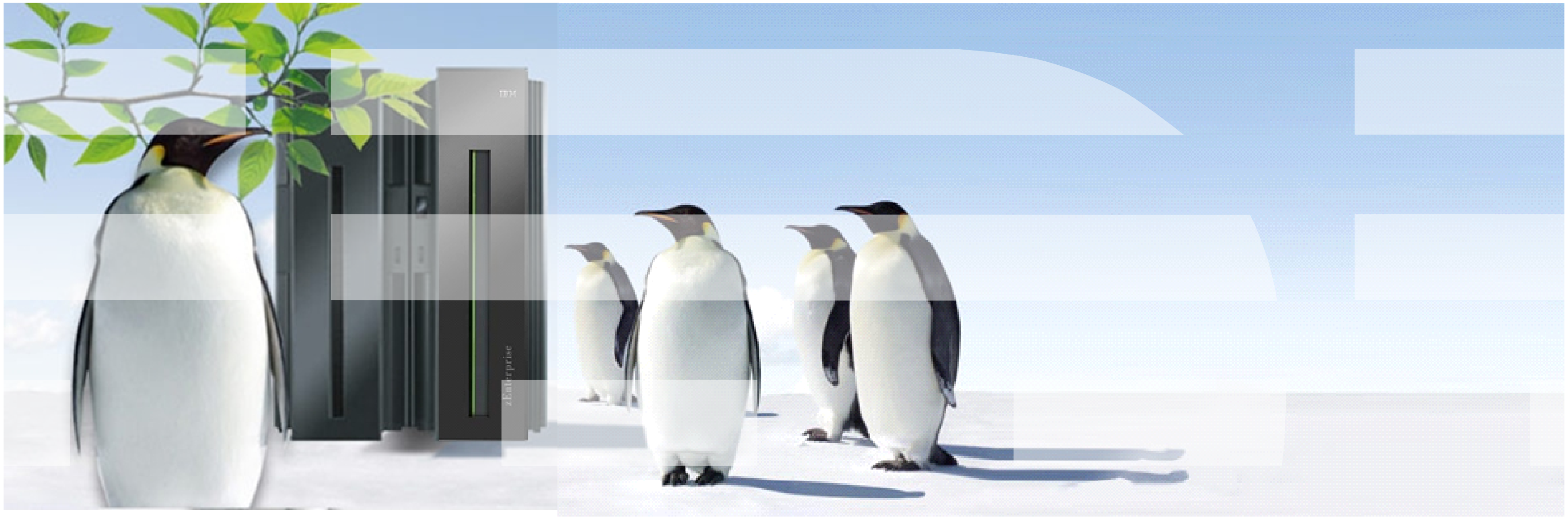
Hendrik Brückner, Linux on System z Kernel Development, brueckner@de.ibm.com

Introduction to the Linux on System z Terminal Server using z/VM IUCV

Discover how terminal connections over z/VM IUCV let you comfortably manage your Linux instances even in emergencies.

Linux on System z Live Virtual Class
Wednesday, November 17. 2010



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

What's on the agenda?

- **How can the terminal server using IUCV help you?**
- **What does an IUCV terminal environment look like?**
  - How can you establish terminal sessions?
  - How can you set up your IUCV terminal environment?

Why do you need a terminal server using z/VM IUCV?

- **Ask yourself**
  - Are you sick and tired of reconfiguring your Linux instances using a line-mode terminal?
  - How often have you struggled using “ed” to correct configuration files?
    - For example, replacing the IP address in the network configuration
  - ***Why not using vi or emacs?***

How can the terminal server using IUCV help you?

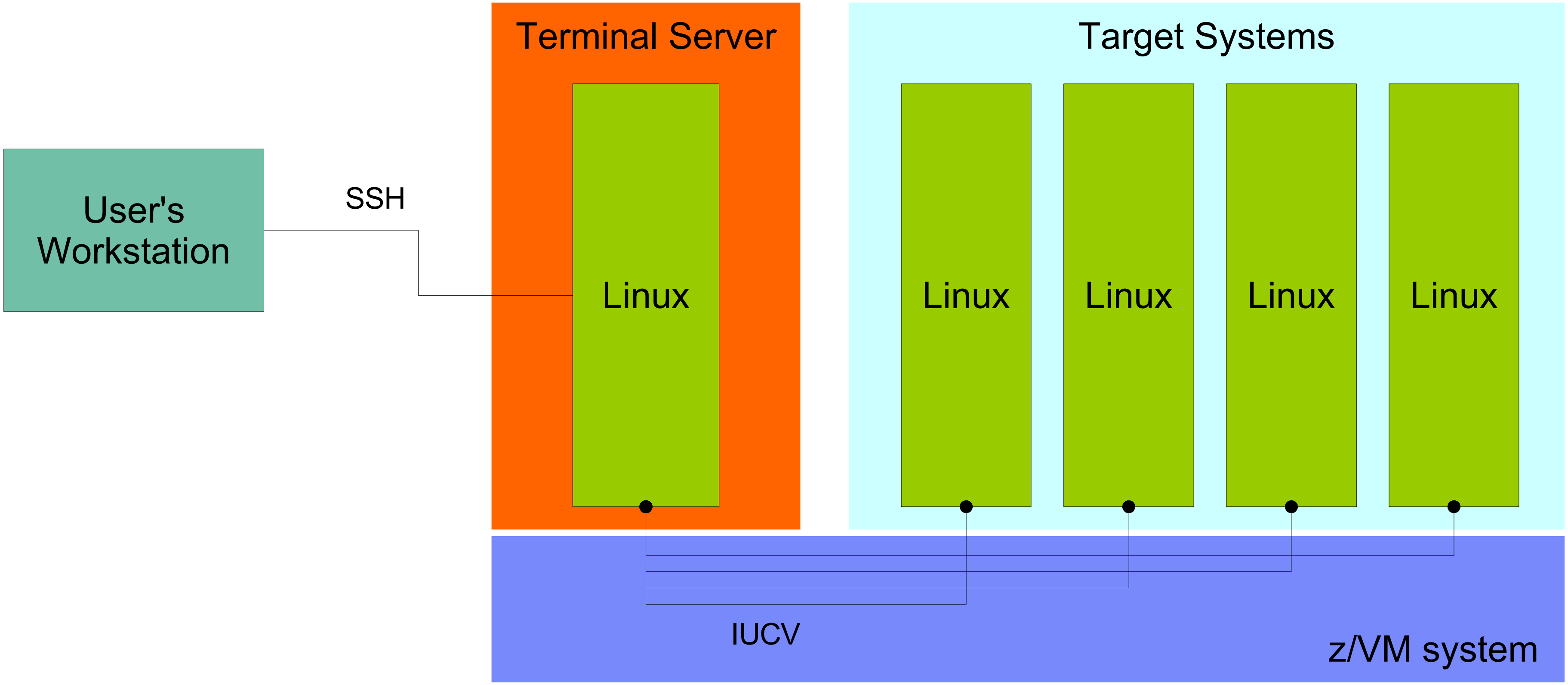
- **Full-screen terminal access to Linux instances on the same z/VM**
- **Access to Linux instances that are not connected to an Internet Protocol (IP) network**
- **Use cases**
  - Provide an alternative terminal access to 3270 and 3215 line-mode terminals
  - Increase availability by providing emergency access if the network for a system fails
  - Centralize access to systems by providing a terminal server environment
  - Heighten security by separating user networks from administrator networks or by isolating sensitive Linux instances from public IP networks

What are terms and what is z/VM IUCV?

- **Linux terminals and consoles**
  - Input/output devices through which users interact with Linux and Linux applications
  - Terminals differ in their modes and capabilities
  - Consoles are output devices which display Linux kernel messages
    - The preferred console displays messages during the boot process
- **Inter-user communication vehicle (IUCV)**
  - A z/VM CP interface for passing data between virtual machines or between CP and a virtual machine
  - The Linux kernel includes IUCV
    - *AF_IUCV – Addressing family for network sockets*
    - *IUCV hypervisor console (HVC) terminal device driver*

What does an IUCV terminal environment look like?

Overview and terminology

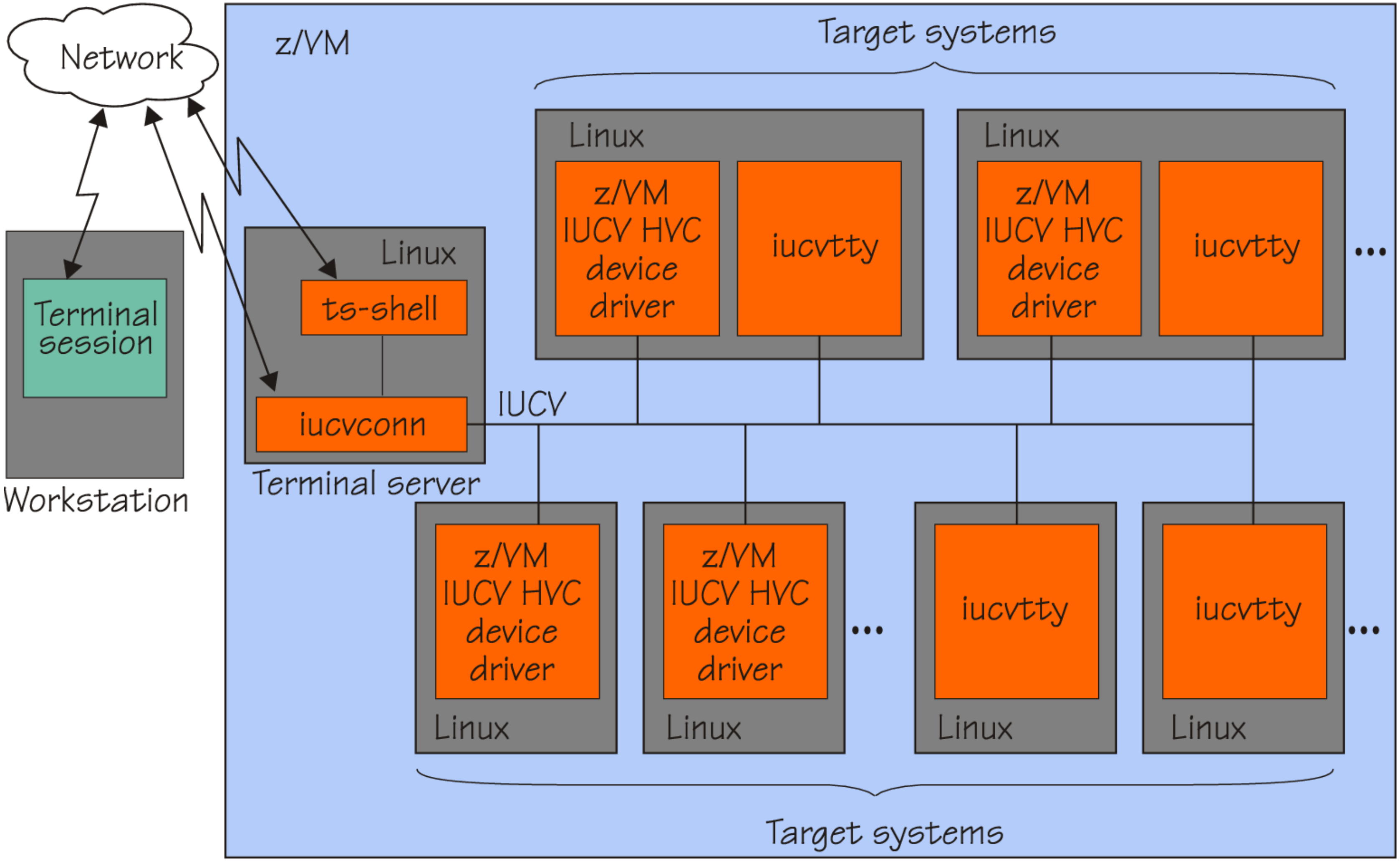


Which programs do you use in an IUCV terminal environment?

- **IUCV terminal programs (s390-tools)**
  - iucvconn – Start terminal connection over IUCV
  - iucvtty – Allow remote logins over IUCV
  - ts-shell – Login shell for setting up a terminal server using IUCV
  - chiucvallow – Restrict access to IUCV HVC terminal devices

- **Terminal access over IUCV is provided by**
  - iucvtty instances
  - IUCV hypervisor console (HVC) device driver (Linux kernel)

What does an IUCV terminal environment look like?



How can you enable a terminal server for iucvconn?

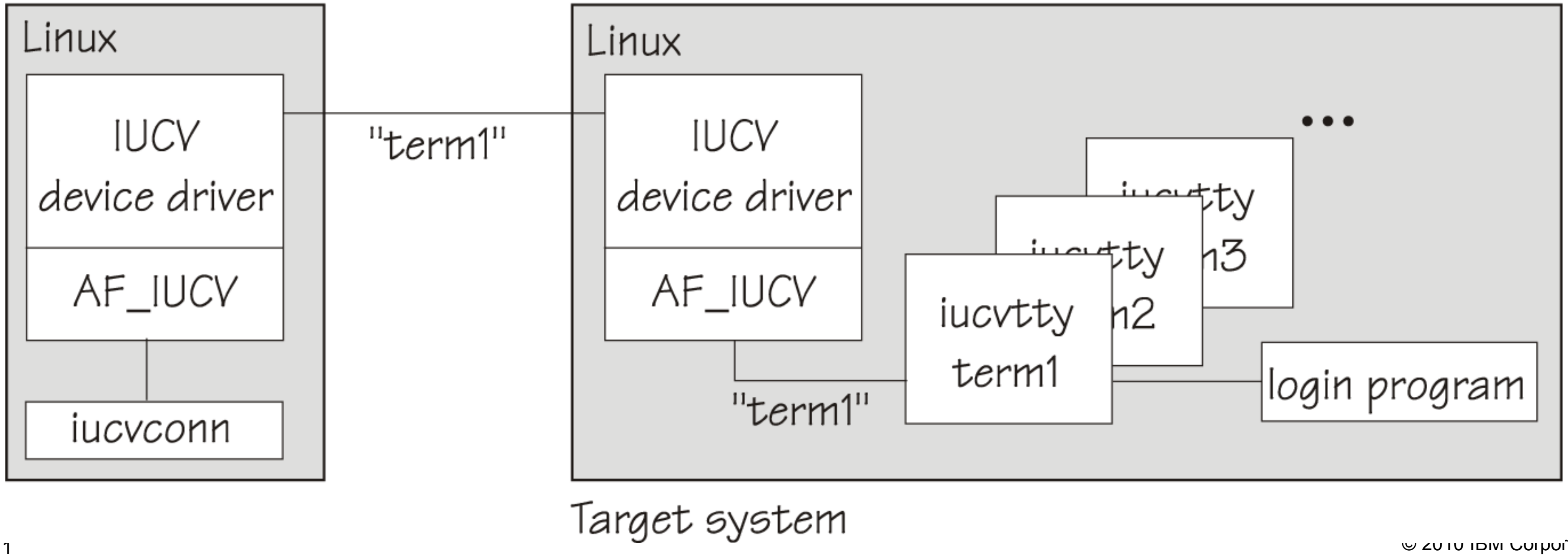
- **Authorizing the z/VM guest virtual machine for IUCV**
  - Adding an IUCV user directory statement, for example, `IUCV ANY`
  - The z/VM user directory for a terminal server might look like:

```
USER LNXTS    XSECRETX 768M 1G G
* General statements
    IPL 0150
    MACH ESA 8
* IUCV authorization
    IUCV ANY
    OPTION MAXCONN 128
* Generic device statements
    CONSOLE 0009 3215 T
    SPOOL 000C 2540 READER *
*   ...
```

How can you establish IUCV terminal sessions?

Establishing terminal sessions to iucvtty instances

- **iucvconn establishes terminal sessions**
  - Socket communication is based on the AF_IUCV address family
  - Addressing is based on the z/VM user ID and a terminal identifier (“`term1`”)
- **iucvtty waits for incoming connections and starts /bin/login to log on users**



How can you set up iucvtty instances on target systems?

1. **Choosing a terminal identifier**
   - For example: `term1`
2. **Enabling user logins**
   - Start the iucvtty program through `/etc/inittab`

```
i1:2345:respawn:/usr/bin/iucvtty term1
```

How do you use iucvconn?

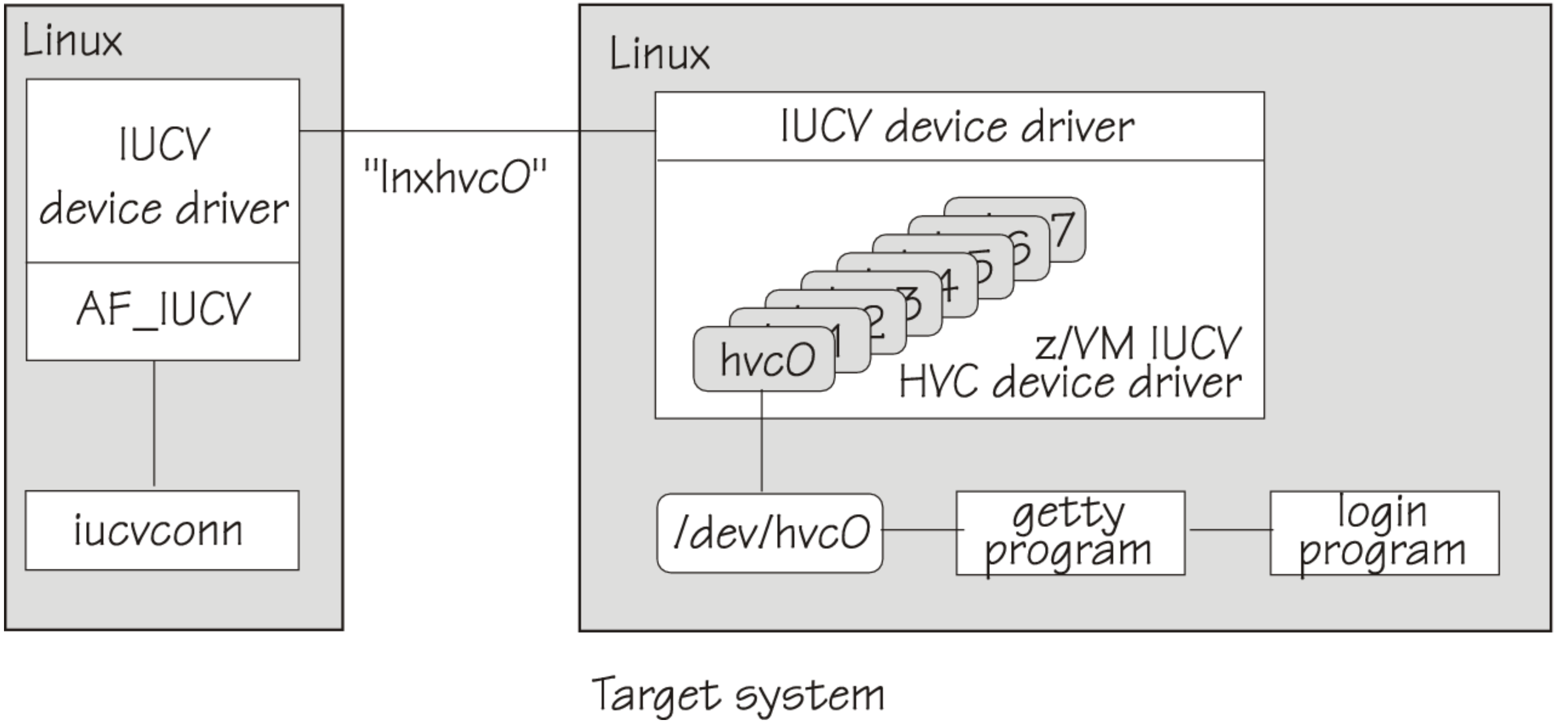
Establishing terminal connections with iucvconn to iucvtty instances

```
brueckh@cetus:~$ ssh brueckner@lnxts
Password:
brueckner@lnxts:~$ iucvconn LINUX005 term1
login: brueckner
Password:
[brueckner@linux005 ~]$ ls
[brueckner@linux005 ~]$ ps
  PID TTY          TIME CMD
 1731 pts/0    00:00:00 bash
 1762 pts/0    00:00:00 ps
[brueckner@linux005 ~]$
```

How can you establish IUCV terminal sessions?

Establishing terminal sessions to HVC terminal devices

- **IUCV HVC device driver provides up to 8 terminal devices (/dev/hvc0 .. /dev/hvc7)**
  - Using the terminal identifiers “`lnxhvc0`” through “`lnxhvc7`”



How can you set up HVC terminal devices on target systems?

1. **Specifying the number of IUCV HVC terminal devices**
   – Set kernel parameter: `hvc_iucv=2`
2. **Enabling user logins**
   – Start a getty program on each terminal through `/etc/inittab`

```
h0:2345:respawn:/sbin/mingetty hvc0
h1:2345:respawn:/sbin/agetty -L 9600 hvc1 xterm
```

3. **Permitting root logins**
   – List HVC terminal devices in `/etc/securetty`
4. **Activating hvc0 to receive Linux kernel messages**
   – Set kernel parameter: `console=hvc0 console=ttyS0`

How do you use iucvconn?

Establishing terminal connections with iucvconn to HVC terminal devices

```
brueckh@cetus:~$ ssh brueckner@lnxts
Password:
brueckner@lnxts:~$ iucvconn LINUX005 lnxhvc0

Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an s390x

linux005 login: root
Password:
Last login: Fri Mar  5 14:31:23 on hvc0
[root@linux005 ~]# ps
  PID TTY          TIME CMD
 4305 hvc0     00:00:00 bash
 4480 hvc0     00:00:00 ps
[root@linux005 ~]# logout
```

What are the differences between iucvtty and IUCV HVC?

Criteria	iucvtty	IUCV HVC device driver
Origin	s390-tools	Linux kernel
Number of terminal instances	> 8	<= 8
Terminal identifiers	variable	fixed
Direct root login	☒	☑
Receiving kernel messages	☒	☑
Acting as preferred console	☒	☑
Restricting access to terminals	☑	☑
Typical use case	administrative actions	emergency actions

What else can you do with iucvconn?

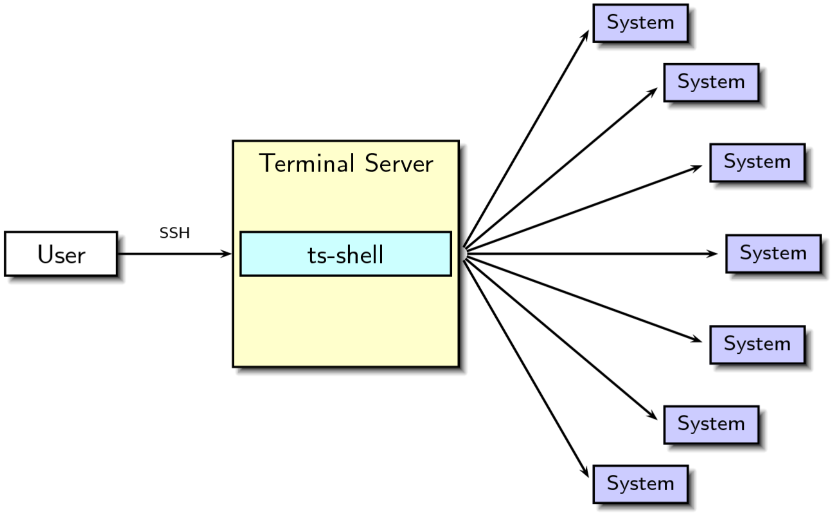
- **Accessing special functions through escape characters**
  - Use Ctrl+_ followed by “d” to disconnect terminal sessions
- **Creating transcripts of terminal sessions with target systems**
  - Writing the terminal data stream to a log file (transcript)
  - Replaying transcripts with realistic output delays

```
brueckh@cetus:~$ ssh brueckner@lnxts
Password:
brueckner@lnxts:~$ iucvconn -s ~/transcripts/linux005 LINUX005 lnxhvc0
[...]

brueckner@lnxts:~$ cd transcripts/
brueckner@lnxts:~/transcripts$ ls -lA
linux005  linux005.info  linux005.timing

brueckner@lnxts:~/transcripts$ scriptreplay linux005.timing linux005
```

What can you do with the terminal server shell, called ts-shell?

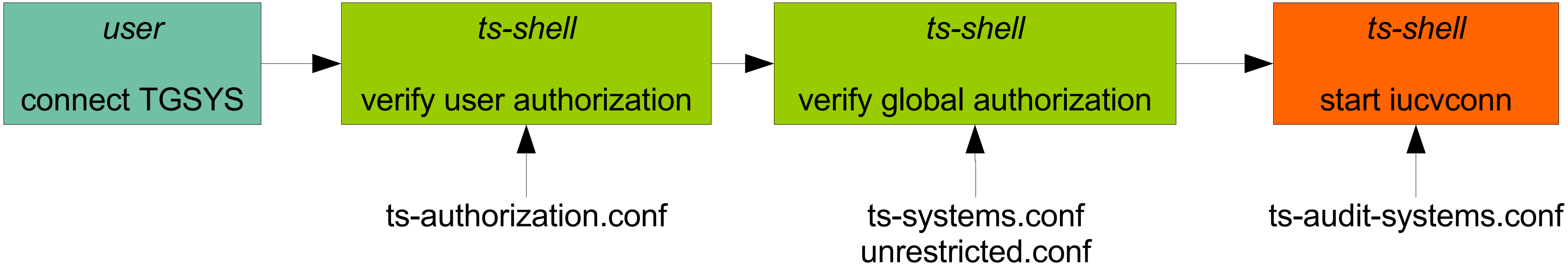
- **ts-shell helps you to**
  - Set up a terminal server to simplify system administration by providing a central access point
  - Authorize users to establish IUCV terminal connections to specific target systems
  - Improve auditing through creating transcripts of terminal sessions with target systems
  - Restrict users from getting access to the terminal server system

- **In a ts-shell session, you can**
  - List your authorizations
  - Establish terminal connections



How does ts-shell perform authorization checks?

- **Before ts-shell connects to a target system, ts-shell verifies that**
  - The user is authorized for the specified target system
  - The target system is included in the global ts-shell authorization
- **ts-shell creates a transcript of a terminal session when**
  - The target system is listed in the audit-systems configuration



How can you authorize users for ts-shell?

1. **Creating a group and a user for ts-shell**

```
groupadd testgrp
useradd -m -s /usr/bin/ts-shell -g ts-shell -G testgrp bob
```

2. **Granting authorization to ts-shell users**

   - Edit `/etc/iucvterm/ts-authorization.conf`

```
@testgrp = list:linux006,linux007,linux008
bob = list:linux005
```

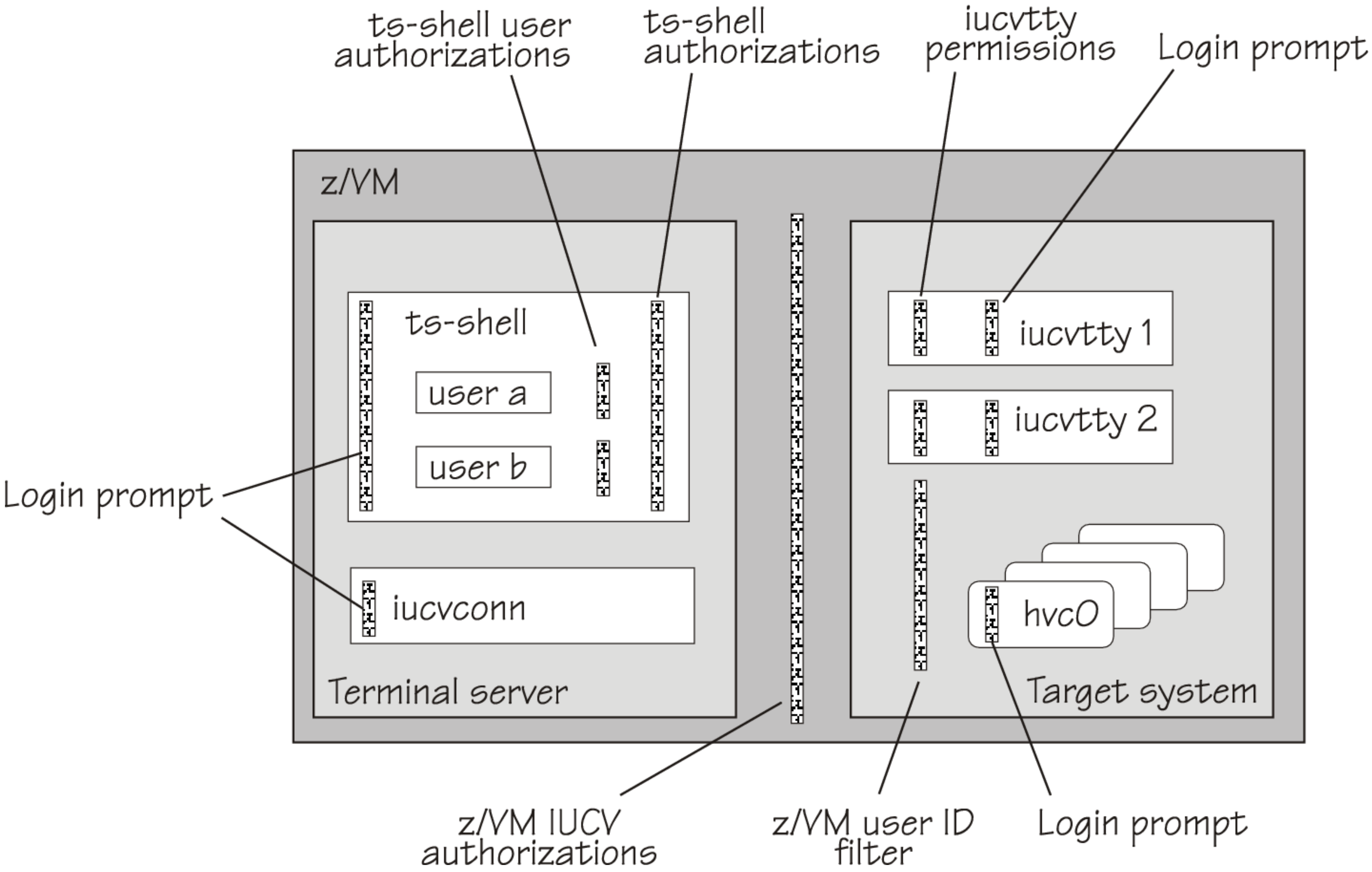
How do you use ts-shell?

Displaying authorizations and establishing terminal connections with ts-shell

```
brueckh@cetus:~$ ssh bob@lnxts
Password:
Last login: Fri Mar  5 12:01:32 2010 from dyn-9-152-212-21
Welcome to the Terminal Server shell.
Type 'help' to get a list of available commands.


bob@ts-shell> list
linux006
linux007
linux008
linux005
bob@ts-shell>
```

```
bob@ts-shell> connect linux005
ts-shell: Connecting to linux005 (terminal identifier: lnxhvc0)...

Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an s390x

linux005 login: root
Password:
Last login: Fri Mar  5 12:02:45 on hvc0
[root@linux005 ~]# ps
  PID TTY          TIME CMD
 1678 hvc0     00:00:00 bash
 1708 hvc0     00:00:00 ps
[root@linux005 ~]# logout
ts-shell: Connection ended
```

How can you secure an IUCV terminal environment?



Conclusion

- **The terminal server using IUCV helps you to**
  - Simplify system administration by providing a central access point
  - Heighten availability by providing emergency access to Linux instances
  - Work more comfortably by using full-screen terminals as alternative to traditional mainframe terminals

Which Linux distributions include the terminal server using IUCV?

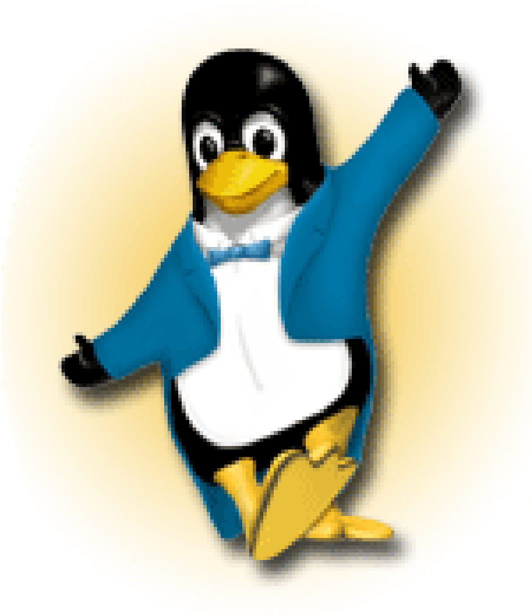
- **Red Hat**
  - Red Hat Enterprise Linux 5 Update 4 or higher
  - Red Hat Enterprise Linux 6
- **Novell**
  - Novell SUSE Linux Enterprise Server 10 Service Pack 3 or higher
  - Novell SUSE Linux Enterprise Server 11 Service Pack 1 or higher
- **“Upstream” availability**
  - Linux kernel 2.6.30 or higher
  - s390-tools 1.8.1 or higher

Where do you get more information?

- **developerWorks**
  - How to Set up a Terminal Server Environment (SC34-2596)
  - Device Drivers, Features, and Commands (SC33-8411)
- **s390-tools package**
  - Man pages for iucvconn(1), iucvtty(1), ts-shell(1), af_iucv(7), and hvc_iucv(9)
  - ts-shell README

Thank you!

Any Questions?



Hendrik Brückner

Linux on System z Development

IBM Deutschland Research
& Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 7031 16-1073
brueckner@de.ibm.com

Backup

THANK YOU!

Your Linux on System z Requirements?

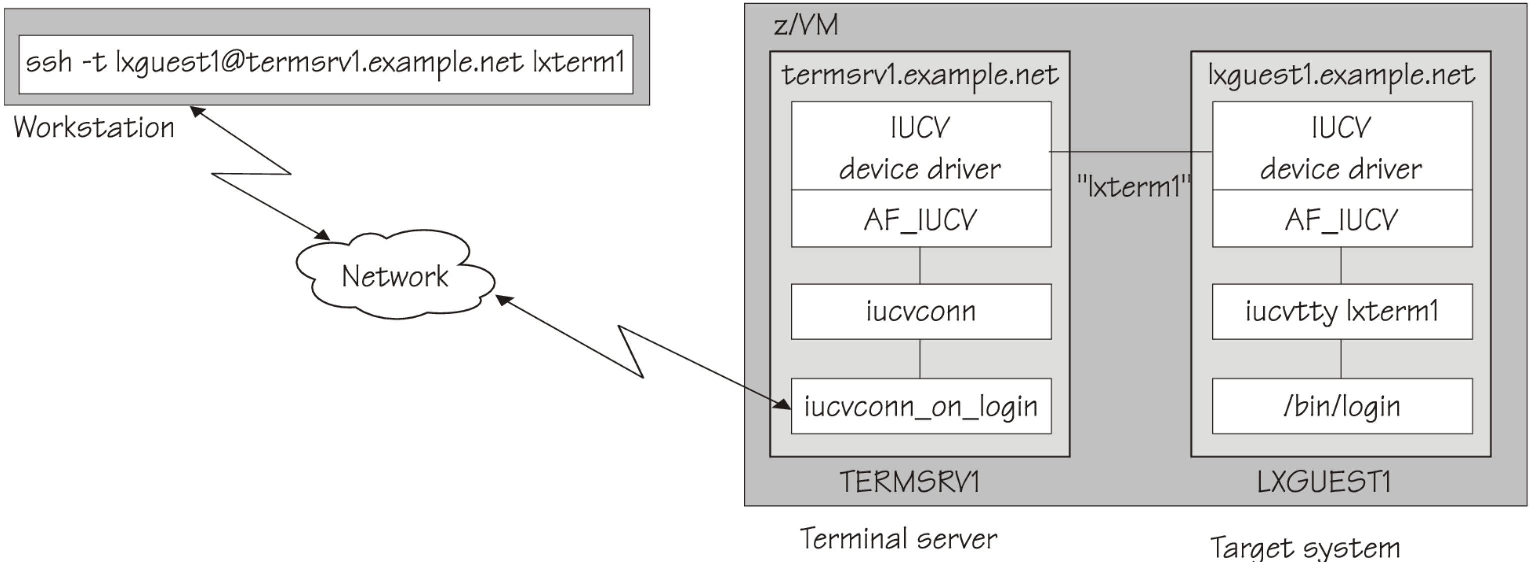
Are you missing a certain feature, functionality, or tool?
We'd love to hear from you!

We will evaluate each request and (hopefully)
develop the additional functionality you need.

Send your input to
brueckner@de.ibm.com

What is iucvconn_on_login?

- **iucvconn_on_login is an alternative login shell for setting up a terminal server**
  - Log in to Linux with a user ID that matches the z/VM user ID of the target system
  - After a successful login, a terminal session is established and the user is prompted to log in to the target system
- **Creating a user for iucvconn_on_login**
  - `useradd -m -s /usr/bin/iucvconn_on_login lxguest1`



What can you do if your terminal setup does not work?

- **What if there is no login prompt when you have connected to a HVC terminal?**
  - Simply press the Return key to reactivate the getty program.
- **What if getty processes respawn too fast on HVC terminal devices?**
  What if getty processes fail to open HVC terminal devices?
  - Check the setting of the `hvc_iucv` kernel parameter in your boot configuration.
  - Run zipl to write the modified boot configuration and reboot.
  - Check `/etc/inittab` or upstart job files to configure only activated HVC terminal devices. Alternatively, use ttyrun to prevent a respawn loop if a terminal is not operational.
- **What if the root user cannot log in on a HVC terminal device?**
  - Check if the HVC terminal device is listed in the `/etc/securetty` file?

How can you restrict access to HVC terminal devices?

Authorizing HVC terminal connections with a z/VM user ID filter

- **The IUCV HVC device driver can accept IUCV connections only from specific terminal servers**
  - Use the `hvc_iucv_allow=` kernel parameter to specify a comma-separated list of authorized z/VM user IDs

- **The chiucvallow command maintains the hvc_iucv_allow setting (filter) at runtime**
  - Display the current z/VM user ID filter with `lsiucvallow` or `chiucvallow -l`
  - Clear the z/VM user ID filter with `chiucvallow -c`
  - Modify the current z/VM user ID filter with `chiucvallow -e`
  - Verify a z/VM user ID filter, which is saved in a file, with `chiucvallow -V`
    - You can also verify ts-shell authorization files that list z/VM user IDs

How do you use the IUCV terminal programs?

- **Using the iucvconn program**
  - To access the first z/VM IUCV HVC terminal on the Linux instance in z/VM guest LNXSYS02

```
$ iucvconn LNXSYS02 lnxhvc0
```

  - To create a transcript of the terminal session to the Linux instance in z/VM guest LNXSYS99

```
$ iucvconn -s ~/transcripts/lnxsys99 LNXSYS99 lnxhvc0
```

- **Using the iucvtty program**
  - To allow remote logins using the terminal identifier “lnxterm”

```
# iucvtty lnxterm
```

  - To access the “lnxterm” terminal on the Linux instance in z/VM guest LNXSYS01

```
$ iucvconn LNXSYS01 lnxterm
```

  - To use /sbin/sulogin instead of /bin/login for terminal identifier “suterm”

```
# iucvtty suterm -- /sbin/sulogin
```

How do you configure Linux instances to provide IUCV terminals?

Examples using inittab

- **Creating new inittab entries for iucvtty instances**

```
t1:2345:respawn:/usr/bin/iucvtty lnxterm
```

- **Creating new inittab entries for IUCV HVC terminal devices**
  - Using the traditional approach:

```
h0:2345:respawn:/sbin/agetty -L 9600 hvc0 xterm
```

  - Using the ttyrun program:

```
h1:2345:respawn:/sbin/ttyrun hvc1 /sbin/agetty -L 9600 %t xterm
```

How do you configure Linux instances to provide IUCV terminals?

Examples using upstart job files

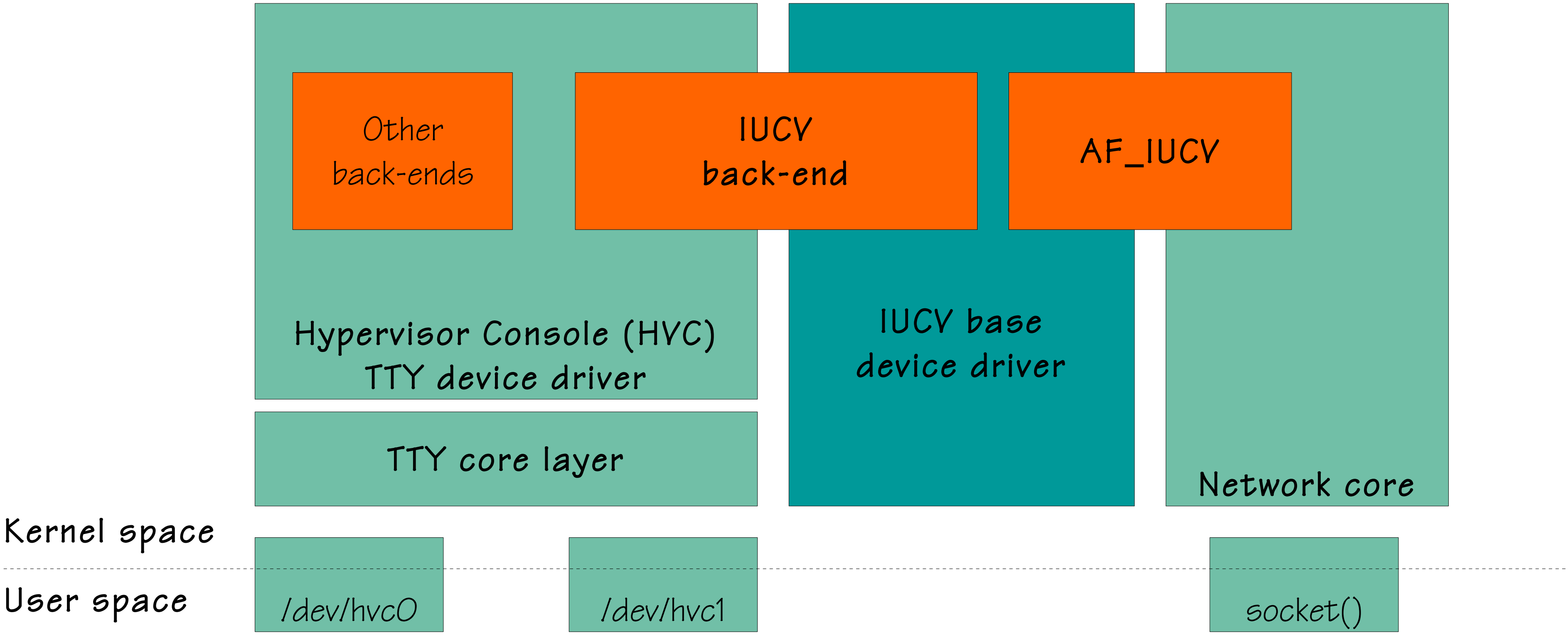
- **Creating a job file for iucvtty**

```
start on runlevel [12345]
stop  on runlevel [06]

respawn
exec /usr/bin/iucvtty lnxterm
console none
```

- **Creating a job file for a HVC terminal device**

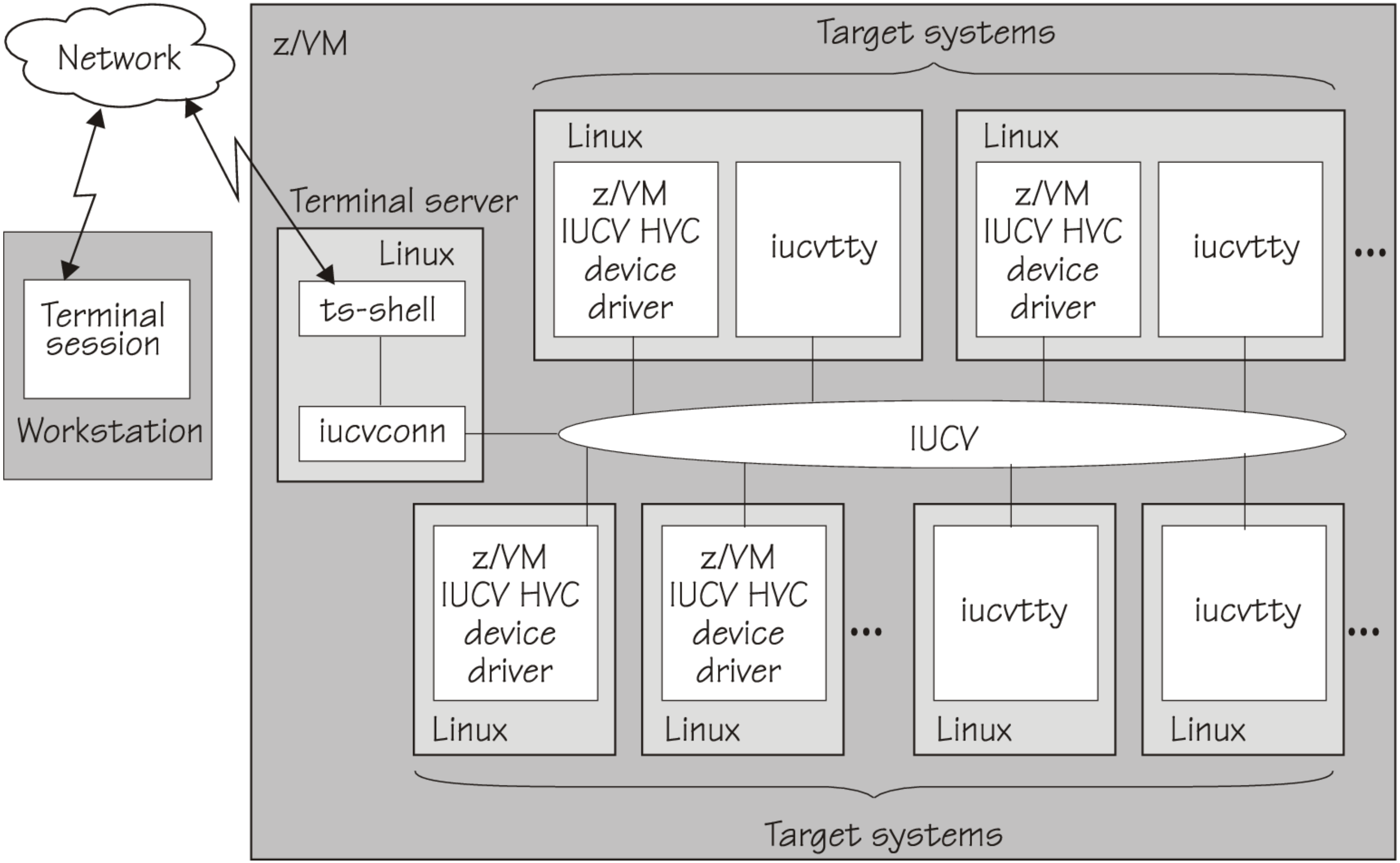
```
start on runlevel [12345]
stop  on runlevel [06]

respawn
normal exit 42
exec /sbin/ttyrun -e 42 hvc0 /sbin/mingetty %t
```

Which Linux kernel components are involved?



What does an IUCV terminal environment look like?

Putting it all together



What are Linux terminals and consoles?

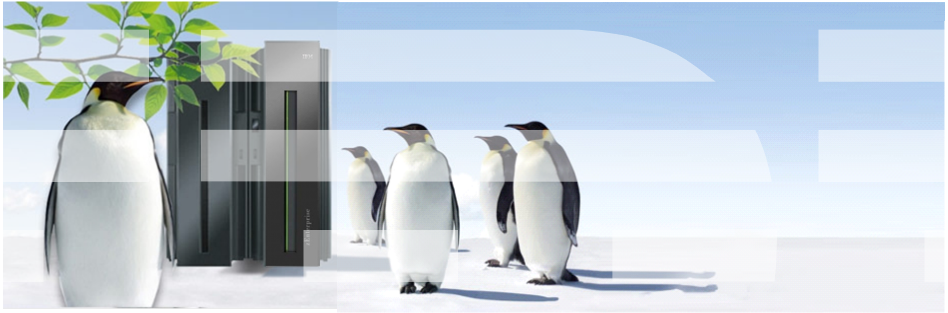
- **Linux terminals**
  - Input/output devices through which users interact with Linux and Linux applications
  - Terminals differ in their modes and capabilities
- **Linux consoles**
  - Consoles are output devices which display Linux kernel messages
  - The preferred console
    - The preferred console is the device that displays messages during the boot process when the 'init' program is called
- **Linux terminal device drivers typically combine terminal and console devices**

Hendrik Brückner, Linux on System z Kernel Development, brueckner@de.ibm.com

Introduction to the Linux on System z Terminal Server using z/VM IUCV

Discover how terminal connections over z/VM IUCV let you comfortably manage your Linux instances even in emergencies.

Linux on System z Live Virtual Class
Wednesday, November 17. 2010



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

What's on the agenda?

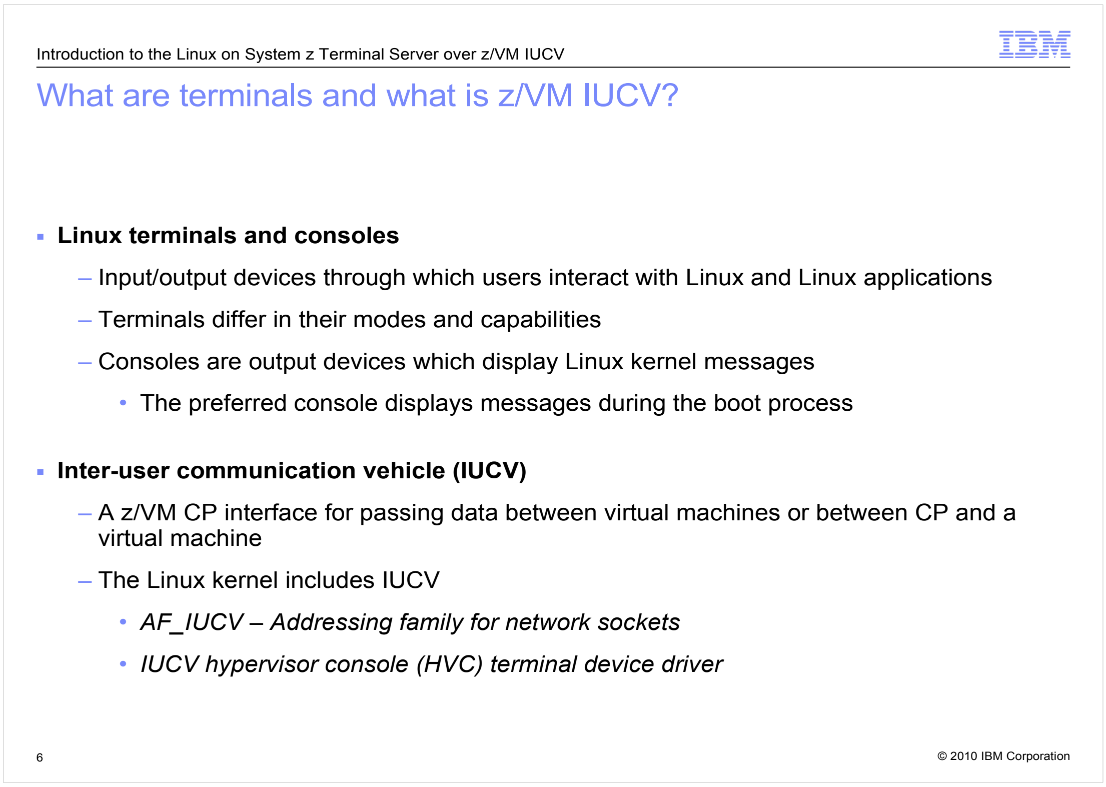
- **How can the terminal server using IUCV help you?**
- **What does an IUCV terminal environment look like?**
  - How can you establish terminal sessions?
  - How can you set up your IUCV terminal environment?

Why do you need a terminal server using z/VM IUCV?

- **Ask yourself**
  - Are you sick and tired of reconfiguring your Linux instances using a line-mode terminal?
  - How often have you struggled using “ed” to correct configuration files?
    - For example, replacing the IP address in the network configuration
  - ***Why not using vi or emacs?***

How can the terminal server using IUCV help you?

- **Full-screen terminal access to Linux instances on the same z/VM**
- **Access to Linux instances that are not connected to an Internet Protocol (IP) network**
- **Use cases**
  - Provide an alternative terminal access to 3270 and 3215 line-mode terminals
  - Increase availability by providing emergency access if the network for a system fails
  - Centralize access to systems by providing a terminal server environment
  - Heighten security by separating user networks from administrator networks or by isolating sensitive Linux instances from public IP networks



What are terminals and what is z/VM IUCV?

- **Linux terminals and consoles**
  - Input/output devices through which users interact with Linux and Linux applications
  - Terminals differ in their modes and capabilities
  - Consoles are output devices which display Linux kernel messages
    - The preferred console displays messages during the boot process
- **Inter-user communication vehicle (IUCV)**
  - A z/VM CP interface for passing data between virtual machines or between CP and a virtual machine
  - The Linux kernel includes IUCV
    - *AF_IUCV – Addressing family for network sockets*
    - *IUCV hypervisor console (HVC) terminal device driver*

Terminal modes:

- Line-mode terminals: 3215
- Block-mode terminals: 3270
- „Full-screen mode“ terminals: allow advanced interactive capabilities

Terminal capabilities:

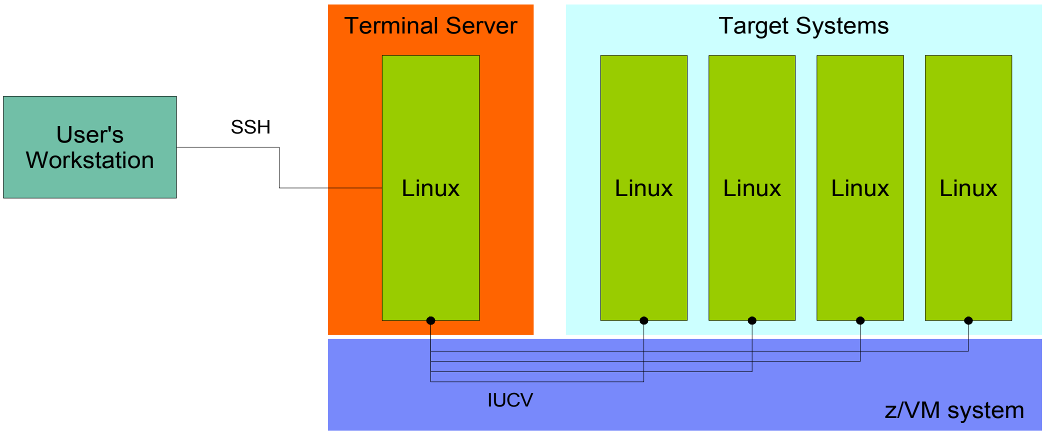
- Specify terminal initialization sequences, screen positioning, colors, ...

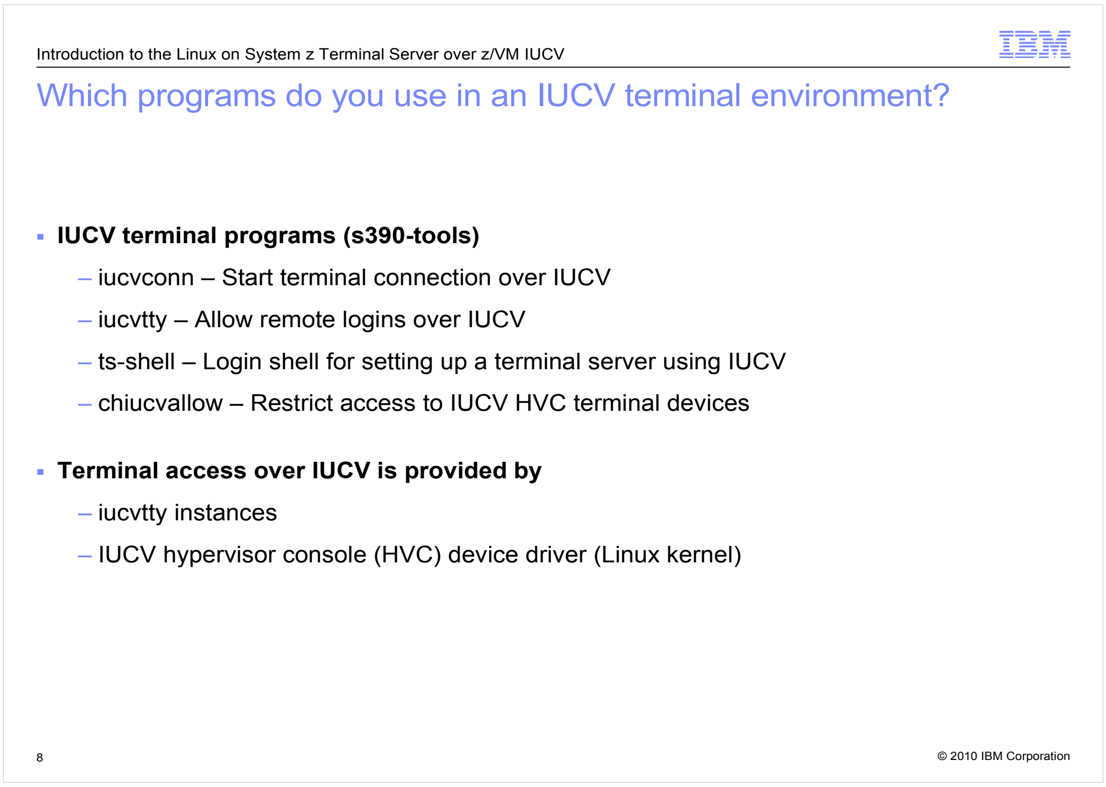
Terminal emulation:

- Today, separate physical terminals are rarities.
- Programs like xterm, kconsole, etc. emulate real terminals.
- The TERM environment variable specifies the set of terminal capabilities. The terminal capabilities are stored in „terminfo“ databases.

What does an IUCV terminal environment look like?

Overview and terminology





Which programs do you use in an IUCV terminal environment?

- **IUCV terminal programs (s390-tools)**
  - iucvconn – Start terminal connection over IUCV
  - iucvtty – Allow remote logins over IUCV
  - ts-shell – Login shell for setting up a terminal server using IUCV
  - chiucvallow – Restrict access to IUCV HVC terminal devices
- **Terminal access over IUCV is provided by**
  - iucvtty instances
  - IUCV hypervisor console (HVC) device driver (Linux kernel)

General:
- The Linux instances must be z/VM guest operating systems of the same z/VM
- Security hints:
  - IUCV authorization for the z/VM guest virtual machine
  - z/VM user ID filter for iucvtty instances and the z/VM IUCV HVC device driver

iucvconn:
- Establishes IUCV connections to either iucvtty instances or HVC terminal devices
- Supports session logging; use „scriptreplay“ to replay transcripts

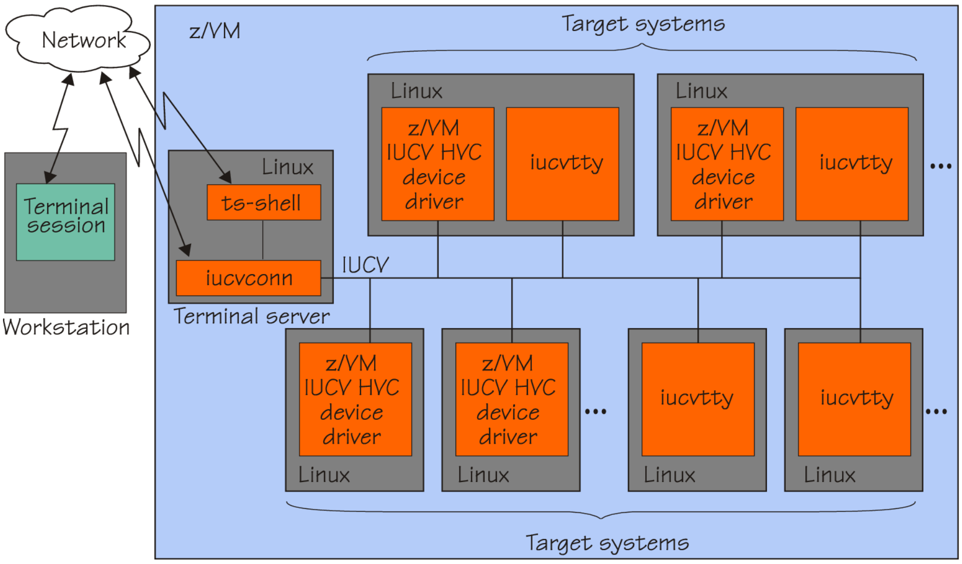
ts-shell:
- Authorizes Linux users based on user names and group memberships for accessing terminals
- Linux users can list the authorizations and access terminals. If a user is authorized to access a terminal, iucvconn is started.

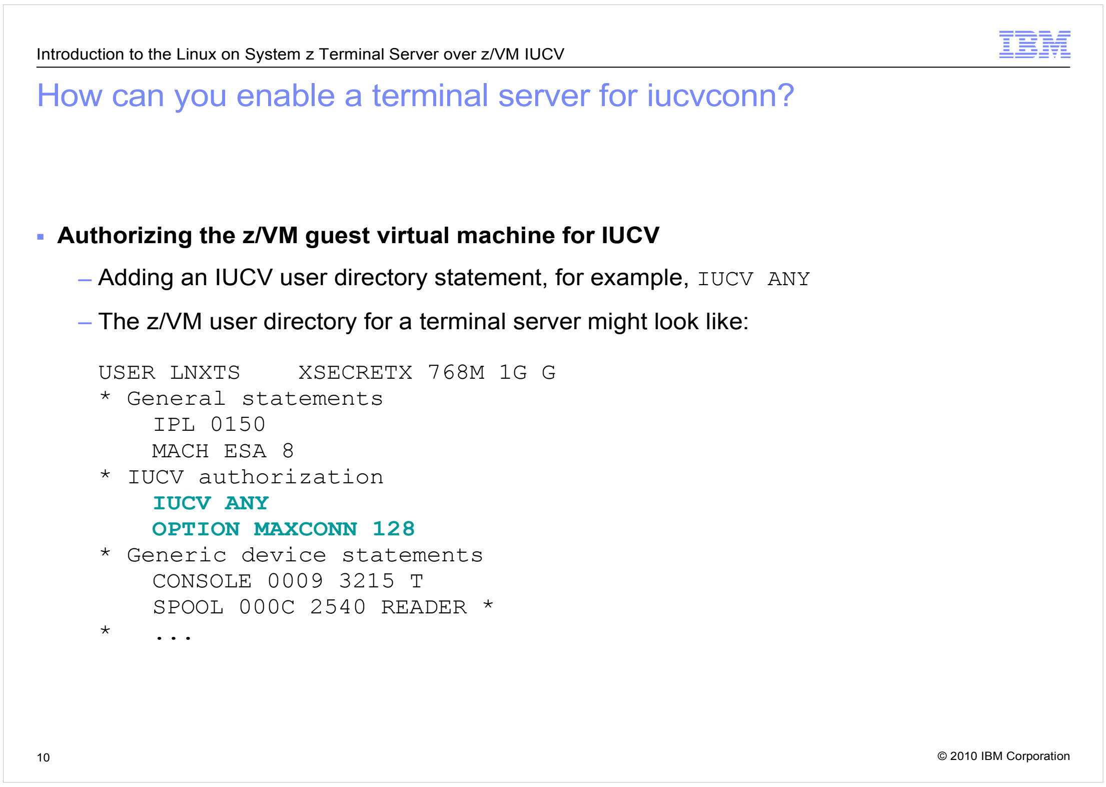
lsiucvallow, chiucvallow:
- List, verify, and change the z/VM user ID filter of the IUCV HVC device driver. The filter specifies the z/VM user IDs that are authorized to access HVC terminal devices.

iucvtty:
- Terminal login on pseudo-terminal devices (pts) using /bin/login
- For security reason, /bin/login does not permit the root user to log in on pseudo-terminal devices.

z/VM IUCV hypervisor console (HVC) device driver (Linux kernel):
- provides up to 8 terminals
- the first terminal can be activated as (preferred) Linux console

What does an IUCV terminal environment look like?





How can you enable a terminal server for iucvconn?

- **Authorizing the z/VM guest virtual machine for IUCV**
  - Adding an IUCV user directory statement, for example, `IUCV ANY`
  - The z/VM user directory for a terminal server might look like:

```
USER LNXTS    XSECRETX 768M 1G G
* General statements
    IPL 0150
    MACH ESA 8
* IUCV authorization
    IUCV ANY
    OPTION MAXCONN 128
* Generic device statements
    CONSOLE 0009 3215 T
    SPOOL 000C 2540 READER *
*   ...
```

Notes

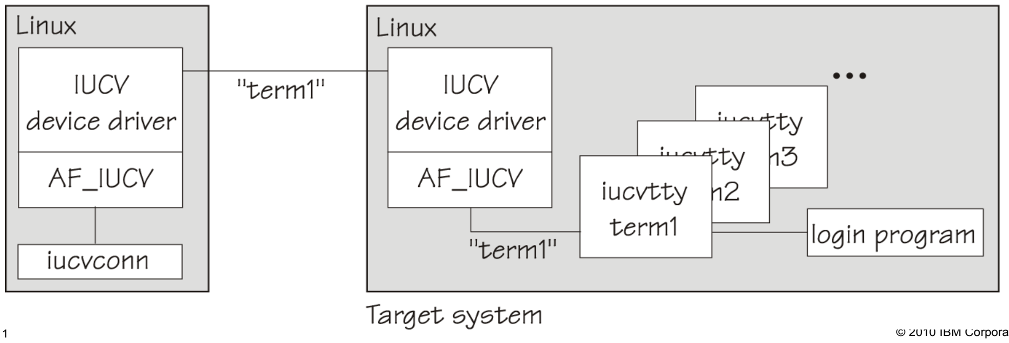
You configure z/VM IUCV authorizations through the IUCV statement in the z/VM user directory. Depending on your needs and security policies you can use different strategies:

- Permit any IUCV connection to a target system:
  IUCV ALLOW
- Permit the terminal server to connect to specific z/VM guest virtual machines:
  IUCV LXGUEST1
  IUCV LXGUEST2
- Permit the terminal server to connect to any z/VM guest virtual machine:
  IUCV ANY

How can you establish IUCV terminal sessions?

Establishing terminal sessions to iucvtty instances

- **iucvconn establishes terminal sessions**
  - Socket communication is based on the AF_IUCV address family
  - Addressing is based on the z/VM user ID and a terminal identifier (“`term1`”)
- **iucvtty waits for incoming connections and starts /bin/login to log on users**



How can you set up iucvtty instances on target systems?

1. **Choosing a terminal identifier**
   - For example: `term1`
2. **Enabling user logins**
   - Start the iucvtty program through `/etc/inittab`

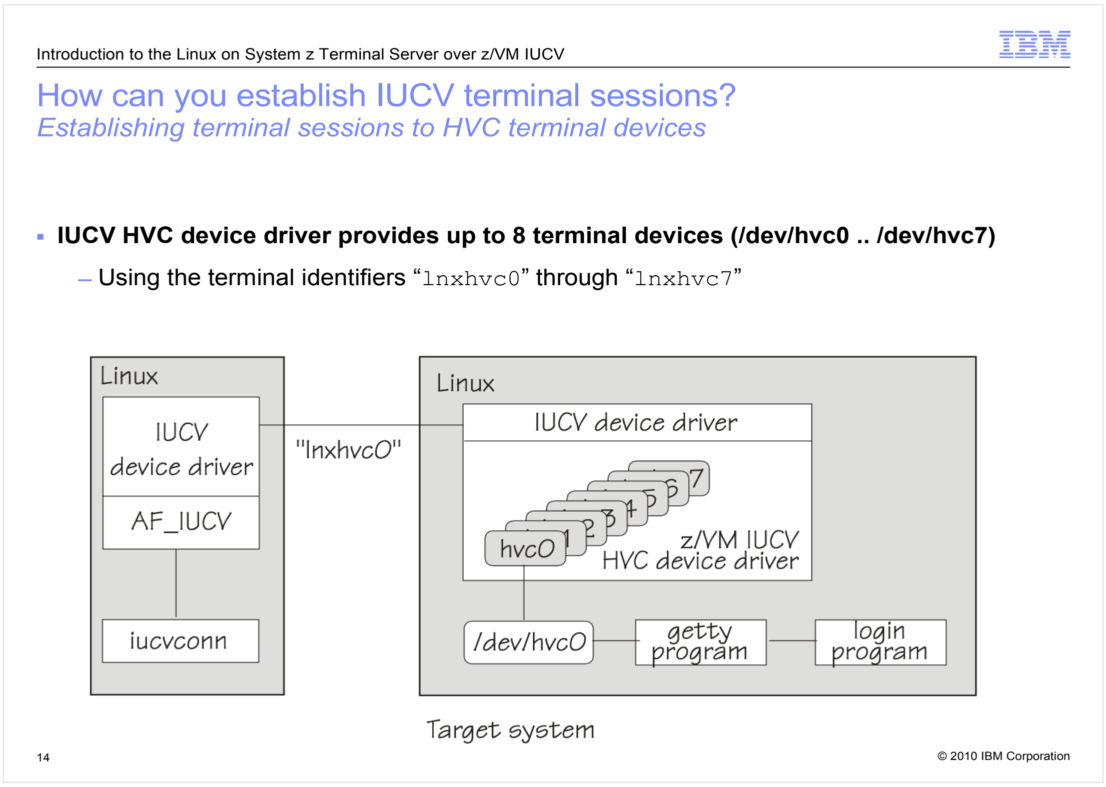
```
i1:2345:respawn:/usr/bin/iucvtty term1
```

How do you use iucvconn?

Establishing terminal connections with iucvconn to iucvtty instances

C

```
brueckh@cetus:~$ ssh brueckner@lnxts
Password:
brueckner@lnxts:~$ iucvconn LINUX005 term1
login: brueckner
Password:
[brueckner@linux005 ~]$ ls
[brueckner@linux005 ~]$ ps
  PID TTY          TIME CMD
 1731 pts/0    00:00:00 bash
 1762 pts/0    00:00:00 ps
[brueckner@linux005 ~]$
```



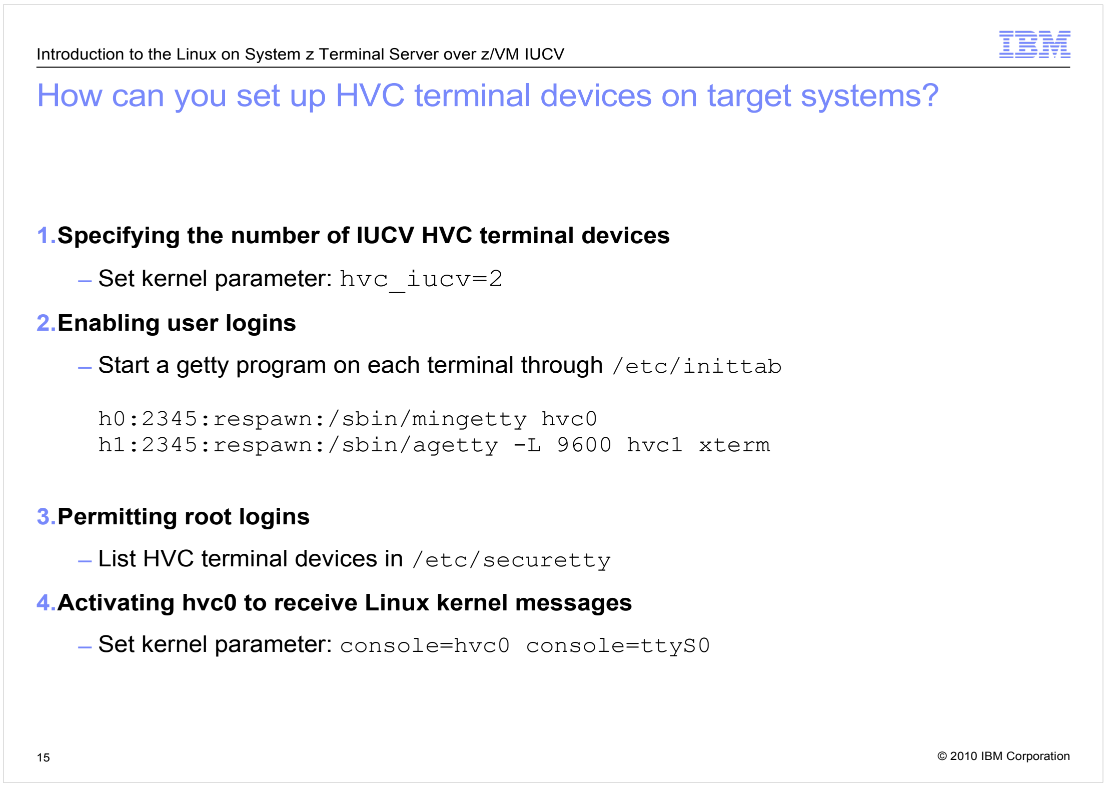
How can you establish IUCV terminal sessions?

Establishing terminal sessions to HVC terminal devices

- **IUCV HVC device driver provides up to 8 terminal devices (/dev/hvc0 .. /dev/hvc7)**
  - Using the terminal identifiers “`lnxhvc0`” through “`lnxhvc7`”

The z/VM IUCV HVC device driver supports the following kernel parameters:

- hvc_iucv=*number* Specifies the number of IUCV HVC terminals (max 8).
- hvc_iucv_allow=*list* Specifies a list of comma-separated z/VM user IDs which are authorized to access. At runtime, the filter can be modified with the *chiucvallow* program.



How can you set up HVC terminal devices on target systems?

1. **Specifying the number of IUCV HVC terminal devices**
   - Set kernel parameter: `hvc_iucv=2`
2. **Enabling user logins**
   - Start a getty program on each terminal through `/etc/inittab`

```
h0:2345:respawn:/sbin/mingetty hvc0
h1:2345:respawn:/sbin/agetty -L 9600 hvc1 xterm
```

3. **Permitting root logins**
   - List HVC terminal devices in `/etc/securetty`
4. **Activating hvc0 to receive Linux kernel messages**
   - Set kernel parameter: `console=hvc0 console=ttyS0`

Setting kernel parameters

- Edit /etc/zipl.conf to add or change the kernel parameters
- Run zipl to write a new boot record with the updated configuration

Activating consoles to receive Linux kernel messages

If you want terminal devices to be activated to receive Linux kernel messages, specify a `console` statement for each of these other devices. The last `console` statement designates the preferred console.

The default preferred console on a Linux on System z system is `ttyS0`. If you specify one or more console parameters and you want to keep `ttyS0` as the preferred console, add a `console` parameter for `ttyS0` as the last console statement.

If you specify `console=hvc0` only, `hvc0` becomes the preferred console. Specify `console=hvc0 console=ttyS0` to receive kernel messages on both devices but keep `ttyS0` as the preferred console.

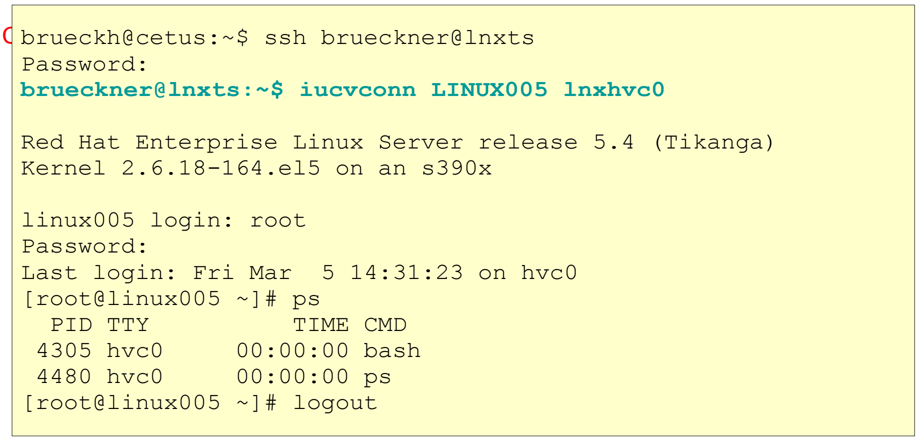
Permitting root logins

The default login program for HVC terminal devices and iucvtty, `/bin/login`, restricts root logins. Root logins are allowed only on devices that are listed in /etc/securetty.

iucvtty uses pseudo-terminal (pts) devices to communicate with the login program. For security reasons, login programs, like `/bin/login`, do not permit root logins on pseudo-terminal devices (see also the man-page for securetty(5)).

How do you use iucvconn?

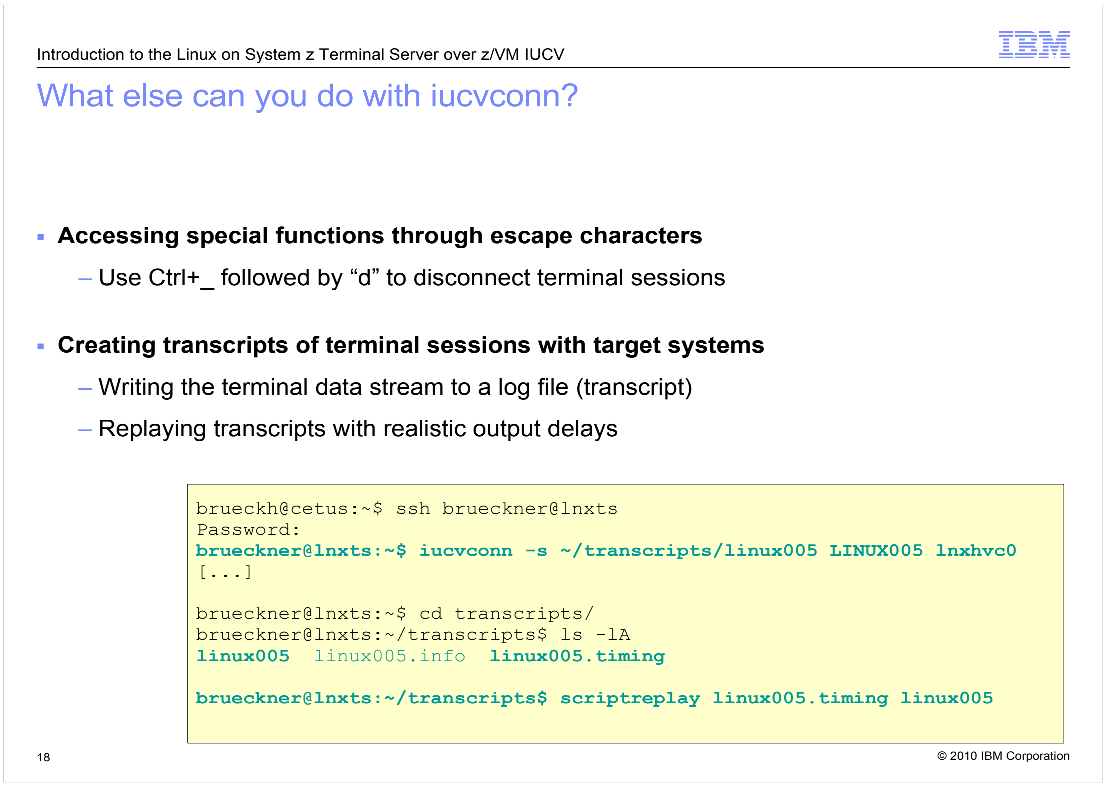
Establishing terminal connections with iucvconn to HVC terminal devices



```
brueckh@cetus:~$ ssh brueckner@lnxts
Password:
brueckner@lnxts:~$ iucvconn LINUX005 lnxhvc0

Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an s390x

linux005 login: root
Password:
Last login: Fri Mar  5 14:31:23 on hvc0
[root@linux005 ~]# ps
  PID TTY          TIME CMD
 4305 hvc0     00:00:00 bash
 4480 hvc0     00:00:00 ps
[root@linux005 ~]# logout
```

What are the differences between iucvtty and IUCV HVC?

Criteria	iucvtty	IUCV HVC device driver
Origin	s390-tools	Linux kernel
Number of terminal instances	> 8	<= 8
Terminal identifiers	variable	fixed
Direct root login	☒	☑
Receiving kernel messages	☒	☑
Acting as preferred console	☒	☑
Restricting access to terminals	☑	☑
Typical use case	administrative actions	emergency actions



What else can you do with iucvconn?

- **Accessing special functions through escape characters**
  - Use Ctrl+_ followed by “d” to disconnect terminal sessions
- **Creating transcripts of terminal sessions with target systems**
  - Writing the terminal data stream to a log file (transcript)
  - Replaying transcripts with realistic output delays

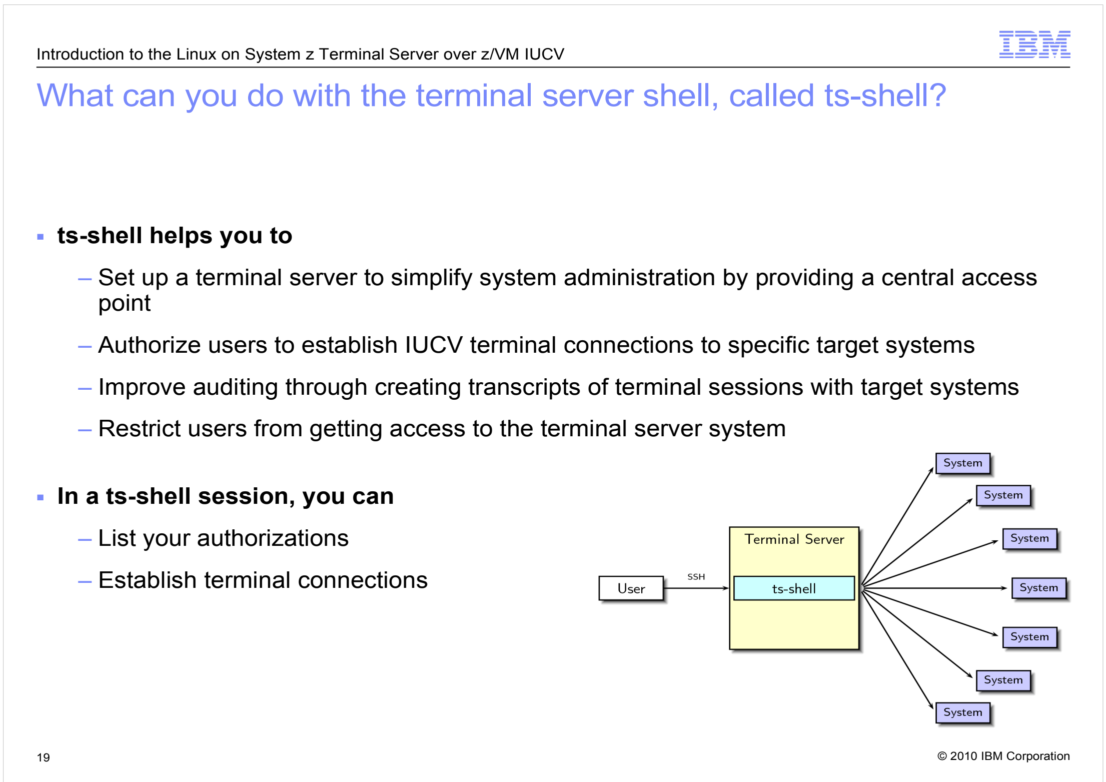
```
brueckh@cetus:~$ ssh brueckner@lnxts
Password:
brueckner@lnxts:~$ iucvconn -s ~/transcripts/linux005 LINUX005 lnxhvc0
[...]

brueckner@lnxts:~$ cd transcripts/
brueckner@lnxts:~/transcripts$ ls -lA
linux005  linux005.info  linux005.timing

brueckner@lnxts:~/transcripts$ scriptreplay linux005.timing linux005
```

Notes on escape characters

- iucvconn uses the underscore (_) character as default escape character
- You can change the escape character with the *-e*, *--escape-char* option
- You can switch off escaping through specifying *-e none*

Notes on terminal session transcripts

- Use the *-s <file>* argument of iucvconn to create a transcript consisting of three files:
  1. *<file>* contains the raw terminal data stream
  2. *<file>.timing* contains timing data for replaying using realistic output delays
  3. *<file>.info* human-readable file containing additional terminal session information
- Replay transcripts with the *scriptreplay* program that is included in the util-linux package.



What can you do with the terminal server shell, called ts-shell?

- **ts-shell helps you to**
  - Set up a terminal server to simplify system administration by providing a central access point
  - Authorize users to establish IUCV terminal connections to specific target systems
  - Improve auditing through creating transcripts of terminal sessions with target systems
  - Restrict users from getting access to the terminal server system
- **In a ts-shell session, you can**
  - List your authorizations
  - Establish terminal connections

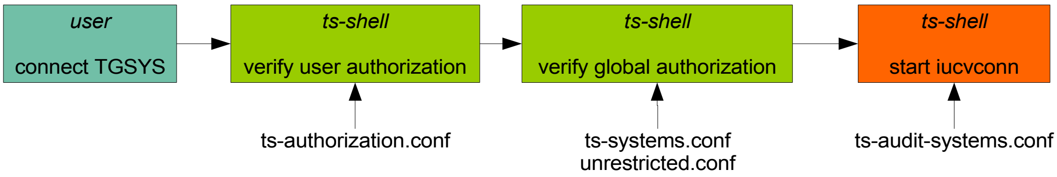
ts-shell commands:

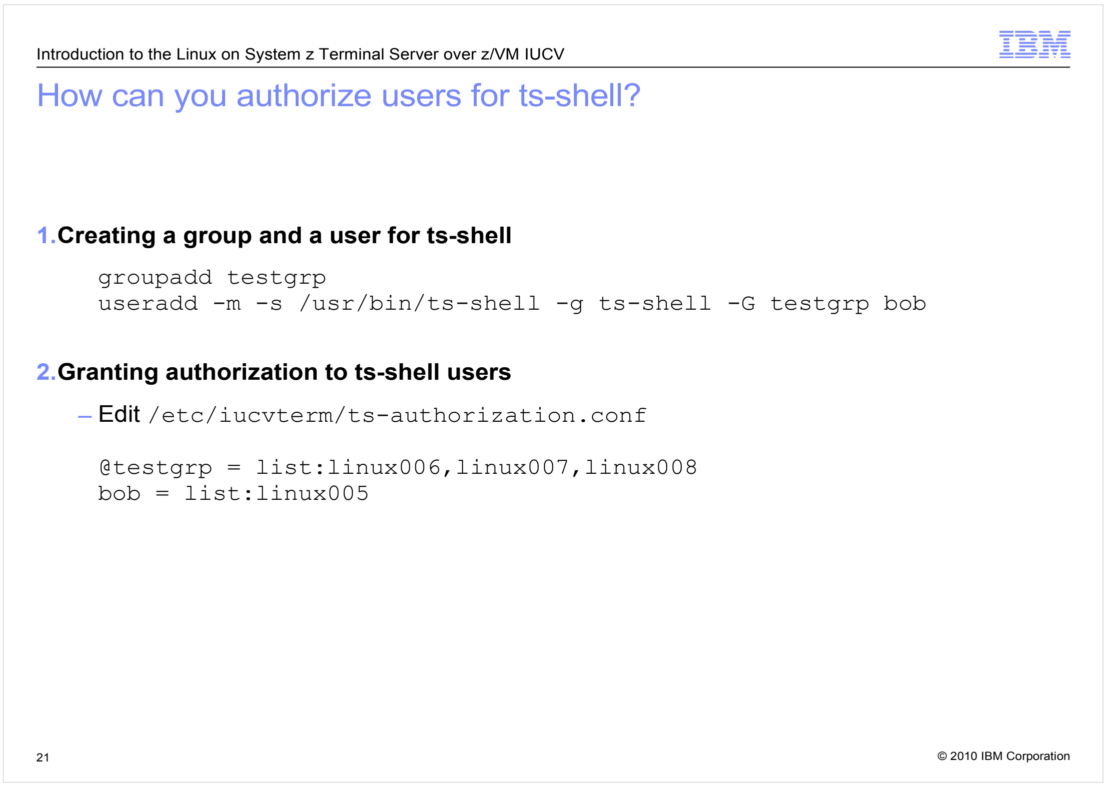
- connect – Establish a terminal session
- list – List authorized target systems
- terminal – Display and set the default terminal ID
- help, version – Display help and version information
- exit, quit – Close terminal server shell session

The ts-shell connect command uses iucvconn to establish terminal sessions. Thus, ts-shell can reuse iucvconn features like creating session transcripts and using escape characters.

How does ts-shell perform authorization checks?

- **Before ts-shell connects to a target system, ts-shell verifies that**
  - The user is authorized for the specified target system
  - The target system is included in the global ts-shell authorization
- **ts-shell creates a transcript of a terminal session when**
  - The target system is listed in the audit-systems configuration





How can you authorize users for ts-shell?

1. **Creating a group and a user for ts-shell**

```
groupadd testgrp
useradd -m -s /usr/bin/ts-shell -g ts-shell -G testgrp bob
```

2. **Granting authorization to ts-shell users**

– Edit `/etc/iucvterm/ts-authorization.conf`

```
@testgrp = list:linux006,linux007,linux008
bob = list:linux005
```

Configuring the ts-shell

The ts-shell program reads its configuration from /etc/iucvterm/ts-shell.conf. The ts-shell.conf file contains settings that specify additional configuration files:

- /etc/iucvterm/unrestricted.conf – Restricting target system connections from ts-shell
- /etc/iucvterm/ts-authorization.conf – Granting authorizations to ts-shell users
- /etc/iucvterm/ts-audit-systems.conf – Configuring session transcripts

Restricting target systems and configuring session transcripts

Both configuration files list z/VM user IDs, each on a separate line.

Granting authorizations to ts-shell users

An authorization statement has the general form: `<users> = <list_type>:<targets>`
`<users>` specifies who is authorized to establish connections. <users> can be an individual Linux user ID or a Linux user group. To distinguish users from groups, groups are prefixed with an at sign (@).
`<list_type>:<targets>` specifies the target systems to which connections are authorized. Target systems can be specified as a comma-separated list (`list:`), in a list file (`file:`), or as a regular expression (`regex:`).

Examples

- The following authorization statement permits user alice to connect to target systems LXGUEST1, LXGUEST3, LXGUEST5, LXGUEST7, and LXGUEST9.

```
alice = list:lxguest1,lxguest3,lxguest5,lxguest7,lxguest9
```

- The following authorization statement permits all users in group `testgrp` to connect to the target systems listed in a file `/etc/iucvterm/auth/testsystems.list`

```
@testgrp = file:/etc/iucvterm/auth/test-systems.list.
```

- The following authorization statement permits user bob to connect to the target systems: LXGUEST0, LXGUEST2, LXGUEST4, LXGUEST6, and LXGUEST8.

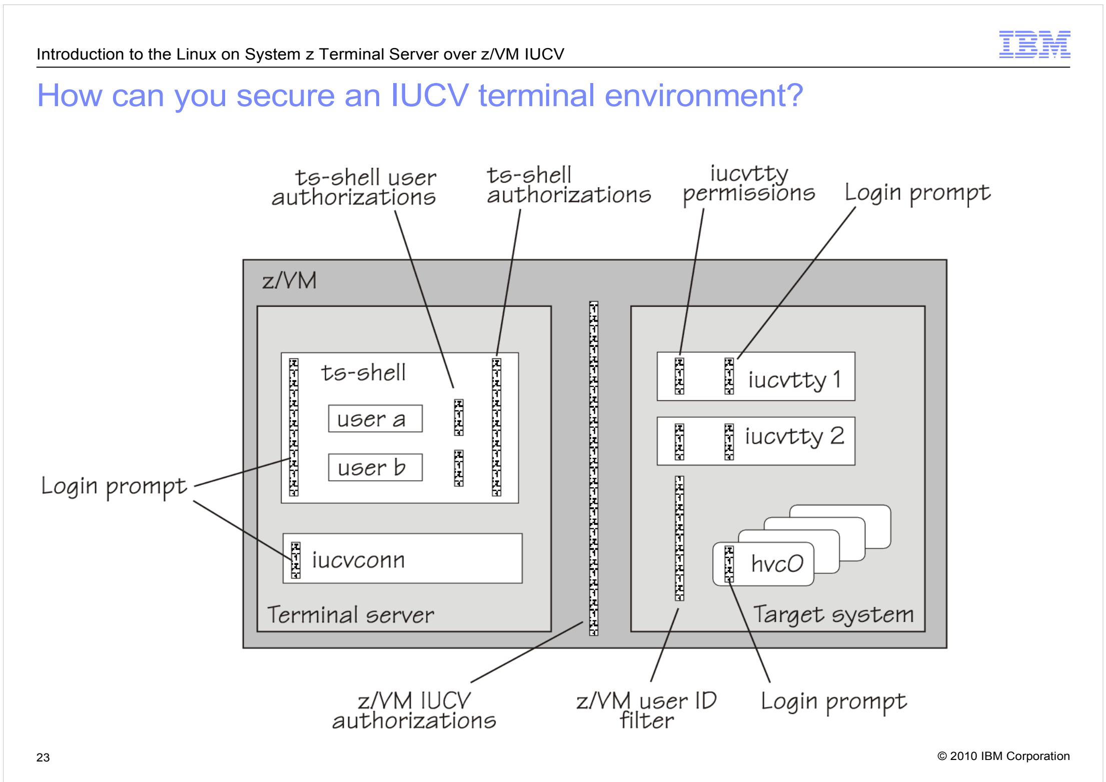
```
bob = regex:lxguest[02468]
```

How do you use ts-shell?

Displaying authorizations and establishing terminal connections with ts-shell

```
brueckh@cetus:~$ ssh bob@lnxts
Password:
Last login: Fri Mar  5 12:01:32 2010 from dyn-9-152-212-21
Welcome to the Terminal Server shell.
Type 'help' to get a list of available commands.


bob@ts-shell> list
linux006
linux007
linux008
linux005
bob@ts-shell>
```

```
bob@ts-shell> connect linux005
ts-shell: Connecting to linux005 (terminal identifier: lnxhvc0)...

Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an s390x

linux005 login: root
Password:
Last login: Fri Mar  5 12:02:45 on hvc0
[root@linux005 ~]# ps
  PID TTY          TIME CMD
 1678 hvc0     00:00:00 bash
 1708 hvc0     00:00:00 ps
[root@linux005 ~]# logout
ts-shell: Connection ended
```



Introduction to the Linux on System z Terminal Server over z/VM IUCV

How can you secure an IUCV terminal environment?

23

Notes

You configure z/VM IUCV authorizations through the IUCV statement in the z/VM user directory. Depending on your needs and security policies you can use different strategies:

- Permit any IUCV connection to a target system:
  IUCV ALLOW
- Permit the terminal server to connect to specific z/VM guest virtual machines:
  IUCV LXGUEST1
  IUCV LXGUEST2
- Permit the terminal server to connect to any z/VM guest virtual machine:
  IUCV ANY

You can restrict access to HVC terminal devices and iucvtty instances on target systems.

- The IUCV HVC device driver includes a z/VM user ID filter which specifies the z/VM user IDs that are allowed to connect. You can specify an initial filter setting through a kernel parameter. Later, you can list, change, or revoke the filter with the *chiucvallow* program.
- The iucvtty program allows you to specify a z/VM user ID filter on the command line.

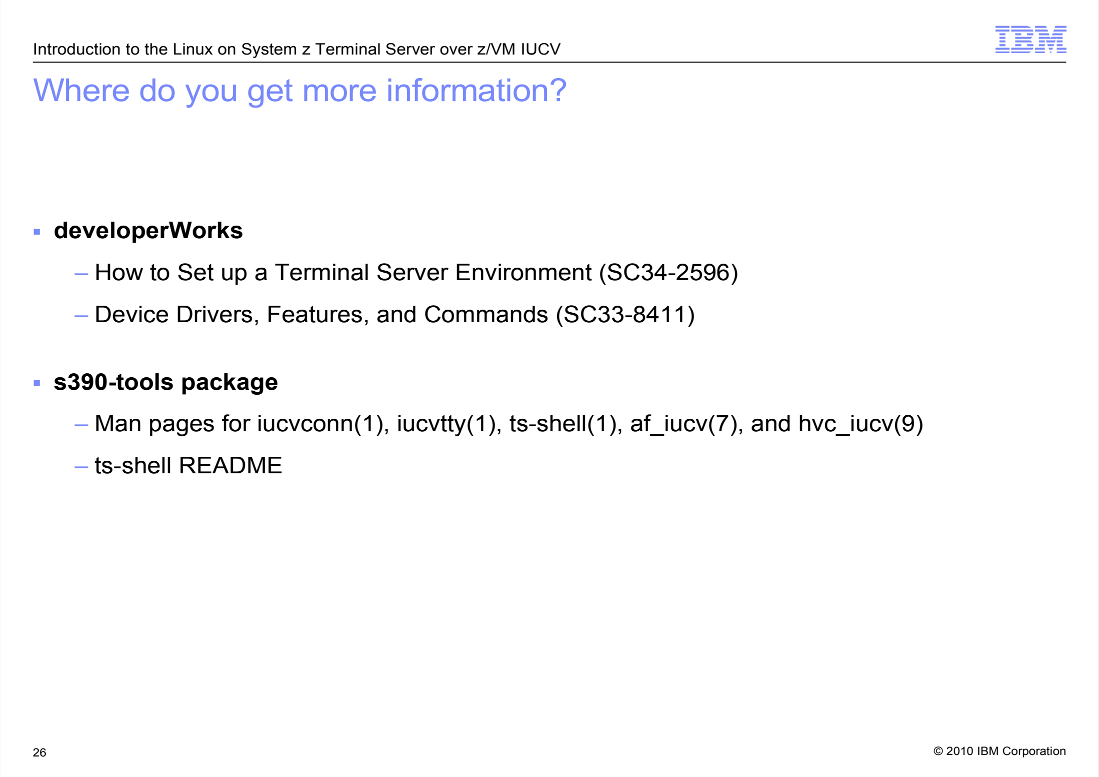
The IUCV terminal programs do not include support for distributed IUCV.

Conclusion

- **The terminal server using IUCV helps you to**
  - Simplify system administration by providing a central access point
  - Heighten availability by providing emergency access to Linux instances
  - Work more comfortably by using full-screen terminals as alternative to traditional mainframe terminals

Which Linux distributions include the terminal server using IUCV?

- **Red Hat**
  - Red Hat Enterprise Linux 5 Update 4 or higher
  - Red Hat Enterprise Linux 6
- **Novell**
  - Novell SUSE Linux Enterprise Server 10 Service Pack 3 or higher
  - Novell SUSE Linux Enterprise Server 11 Service Pack 1 or higher
- **“Upstream” availability**
  - Linux kernel 2.6.30 or higher
  - s390-tools 1.8.1 or higher



Where do you get more information?

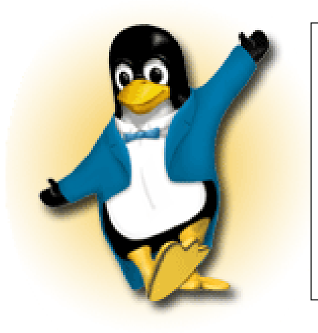
- **developerWorks**
  - How to Set up a Terminal Server Environment (SC34-2596)
  - Device Drivers, Features, and Commands (SC33-8411)
- **s390-tools package**
  - Man pages for iucvconn(1), iucvtty(1), ts-shell(1), af_iucv(7), and hvc_iucv(9)
  - ts-shell README

http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html

Thank you!

- Click to add an outline

Any Questions?



Hendrik Brückner

Linux on System z Development

IBM Deutschland Research
& Development GmbH
Schönaicher Strasse 220
71032 Böblingen, Germany

Phone +49 7031 16-1073
brueckner@de.ibm.com

Backup

- Click to add an outline

Click to add title

THANK YOU!

- Click to add an outline

Your Linux on System z Requirements?

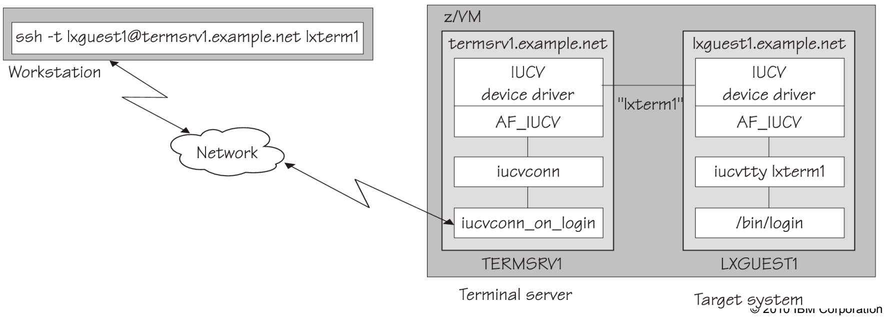
Are you missing a certain feature, functionality, or tool?
We'd love to hear from you!

We will evaluate each request and (hopefully)
develop the additional functionality you need.

Send your input to
brueckner@de.ibm.com

What is iucvconn_on_login?

- **iucvconn_on_login is an alternative login shell for setting up a terminal server**
  - Log in to Linux with a user ID that matches the z/VM user ID of the target system
  - After a successful login, a terminal session is established and the user is prompted to log in to the target system
- **Creating a user for iucvconn_on_login**
  - `useradd -m -s /usr/bin/iucvconn_on_login lxguest1`



What can you do if your terminal setup does not work?

- **What if there is no login prompt when you have connected to a HVC terminal?**
  - Simply press the Return key to reactivate the getty program.
- **What if getty processes respawn too fast on HVC terminal devices?**
  What if getty processes fail to open HVC terminal devices?
  - Check the setting of the `hvc_iucv` kernel parameter in your boot configuration.
  - Run zipl to write the modified boot configuration and reboot.
  - Check `/etc/inittab` or upstart job files to configure only activated HVC terminal devices. Alternatively, use ttyrun to prevent a respawn loop if a terminal is not operational.
- **What if the root user cannot log in on a HVC terminal device?**
  - Check if the HVC terminal device is listed in the `/etc/securetty` file?

How can you restrict access to HVC terminal devices?

Authorizing HVC terminal connections with a z/VM user ID filter

- **The IUCV HVC device driver can accept IUCV connections only from specific terminal servers**
  - Use the `hvc_iucv_allow=` kernel parameter to specify a comma-separated list of authorized z/VM user IDs

- **The chiucvallow command maintains the hvc_iucv_allow setting (filter) at runtime**
  - Display the current z/VM user ID filter with `lsiucvallow` or `chiucvallow -l`
  - Clear the z/VM user ID filter with `chiucvallow -c`
  - Modify the current z/VM user ID filter with `chiucvallow -e`
  - Verify a z/VM user ID filter, which is saved in a file, with `chiucvallow -V`
    - You can also verify ts-shell authorization files that list z/VM user IDs

How do you use the IUCV terminal programs?

- **Using the iucvconn program**
  - To access the first z/VM IUCV HVC terminal on the Linux instance in z/VM guest LNXSYS02

    ```
    $ iucvconn LNXSYS02 lnxhvc0
    ```

  - To create a transcript of the terminal session to the Linux instance in z/VM guest LNXSYS99

    ```
    $ iucvconn -s ~/transcripts/lnxsys99 LNXSYS99 lnxhvc0
    ```

- **Using the iucvtty program**
  - To allow remote logins using the terminal identifier “lnxterm”

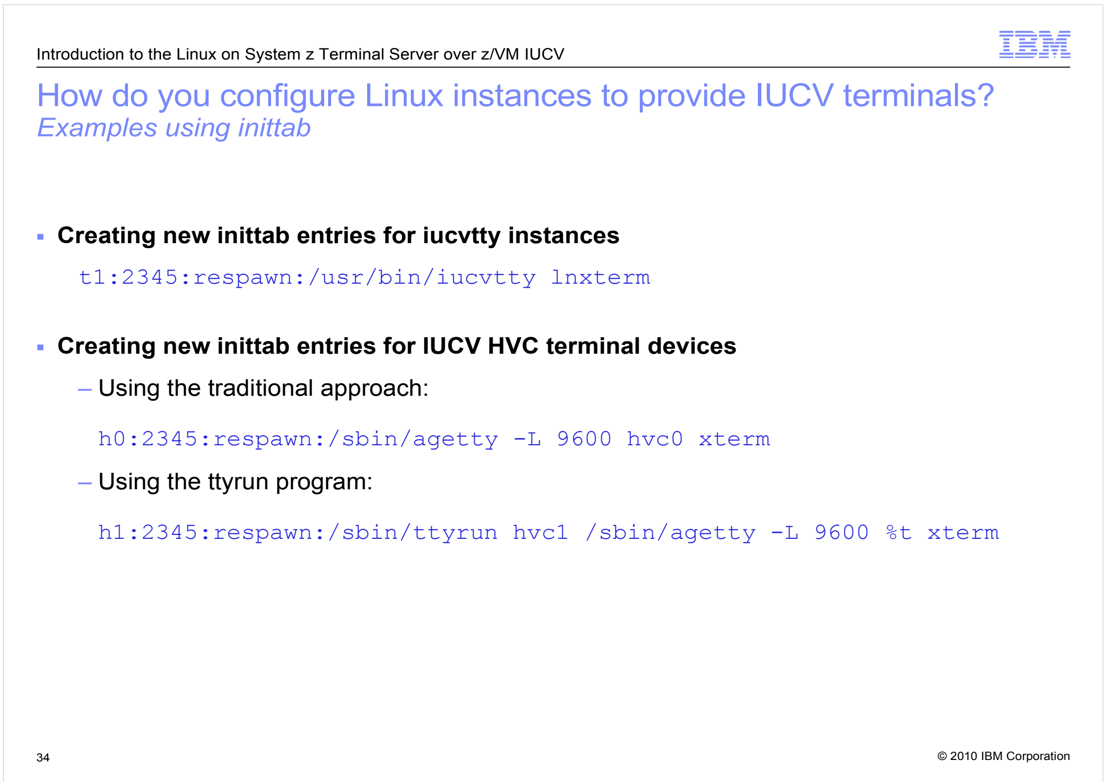
    ```
    # iucvtty lnxterm
    ```

  - To access the “lnxterm” terminal on the Linux instance in z/VM guest LNXSYS01

    ```
    $ iucvconn LNXSYS01 lnxterm
    ```

  - To use /sbin/sulogin instead of /bin/login for terminal identifier “suterm”

    ```
    # iucvtty suterm -- /sbin/sulogin
    ```



How do you configure Linux instances to provide IUCV terminals?

Examples using inittab

- **Creating new inittab entries for iucvtty instances**

```
t1:2345:respawn:/usr/bin/iucvtty lnxterm
```

- **Creating new inittab entries for IUCV HVC terminal devices**
  - Using the traditional approach:

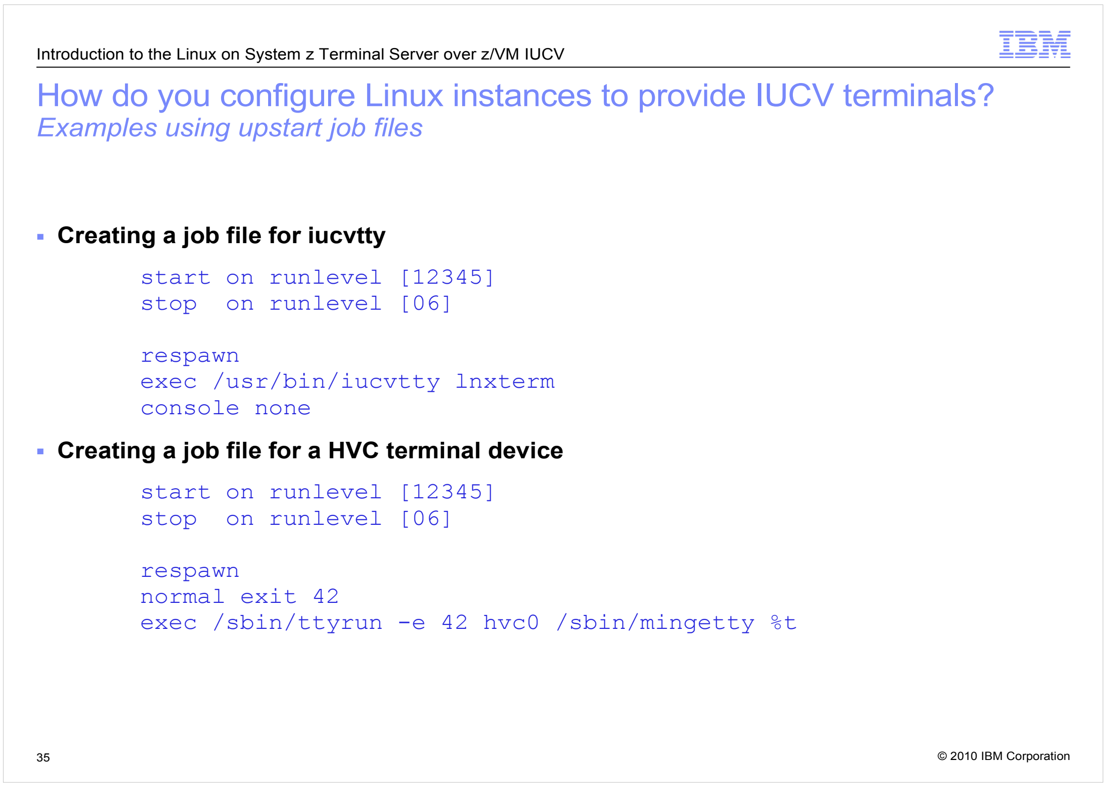
```
h0:2345:respawn:/sbin/agetty -L 9600 hvc0 xterm
```

  - Using the ttyrun program:

```
h1:2345:respawn:/sbin/ttyrun hvc1 /sbin/agetty -L 9600 %t xterm
```

Notes on ttyrun

The ttyrun program is typically started during system initialization and is used to prevent respawn through the init(8) program when a terminal is not available.



How do you configure Linux instances to provide IUCV terminals?

Examples using upstart job files

- **Creating a job file for iucvtty**

```
start on runlevel [12345]
stop  on runlevel [06]

respawn
exec /usr/bin/iucvtty lnxterm
console none
```

- **Creating a job file for a HVC terminal device**

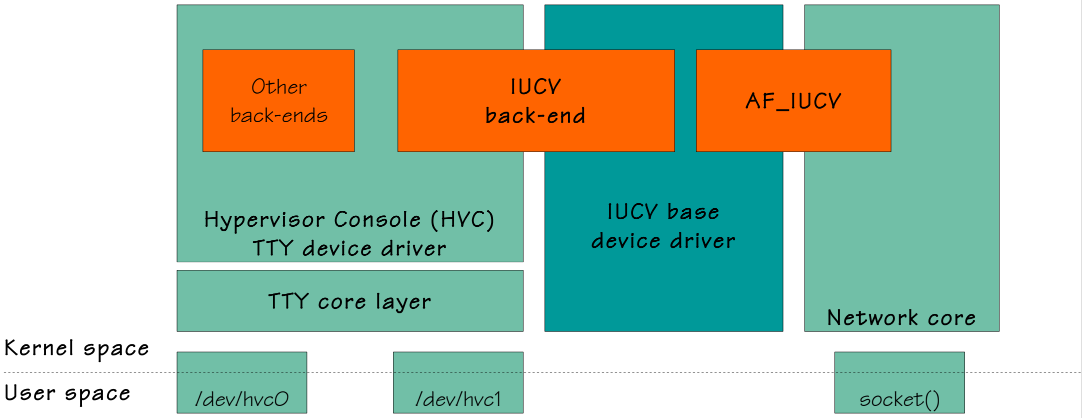
```
start on runlevel [12345]
stop  on runlevel [06]

respawn
normal exit 42
exec /sbin/ttyrun -e 42 hvc0 /sbin/mingetty %t
```

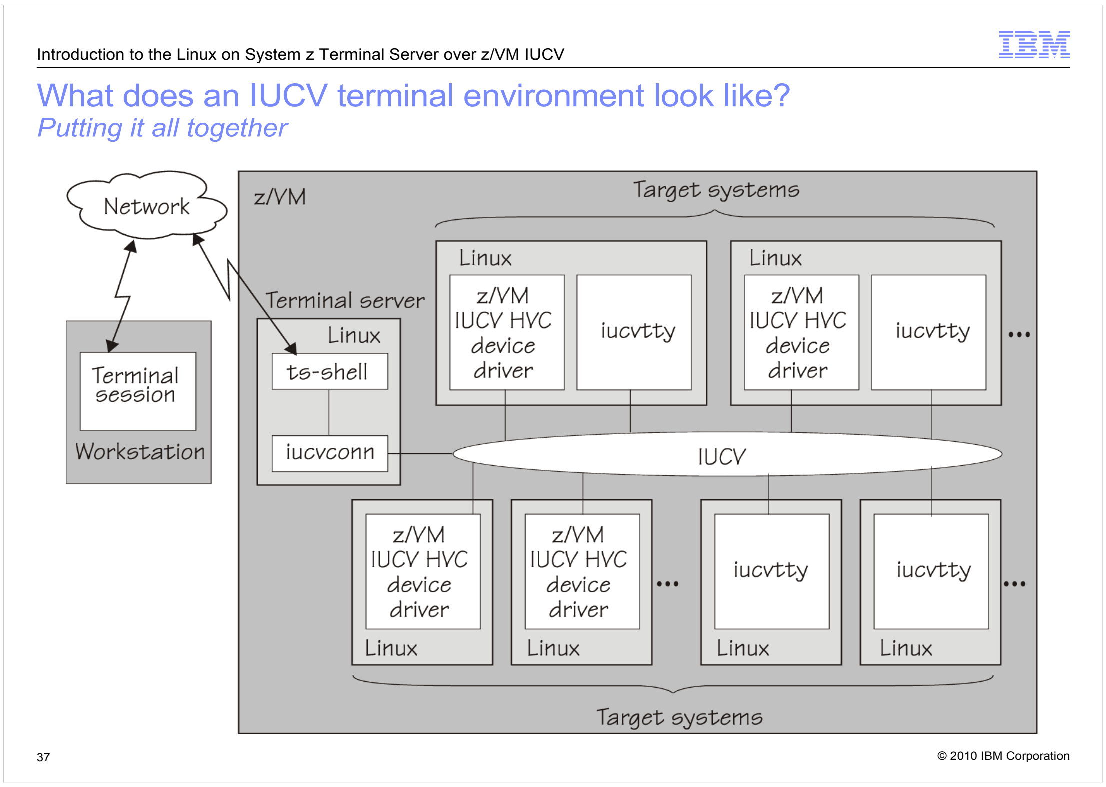
Notes on ttyrun

The ttyrun program is typically started during system initialization and is used to prevent a respawn through the init(8) program when a terminal is not available.

Which Linux kernel components are involved?

- Click to add an outline





What does an IUCV terminal environment look like?

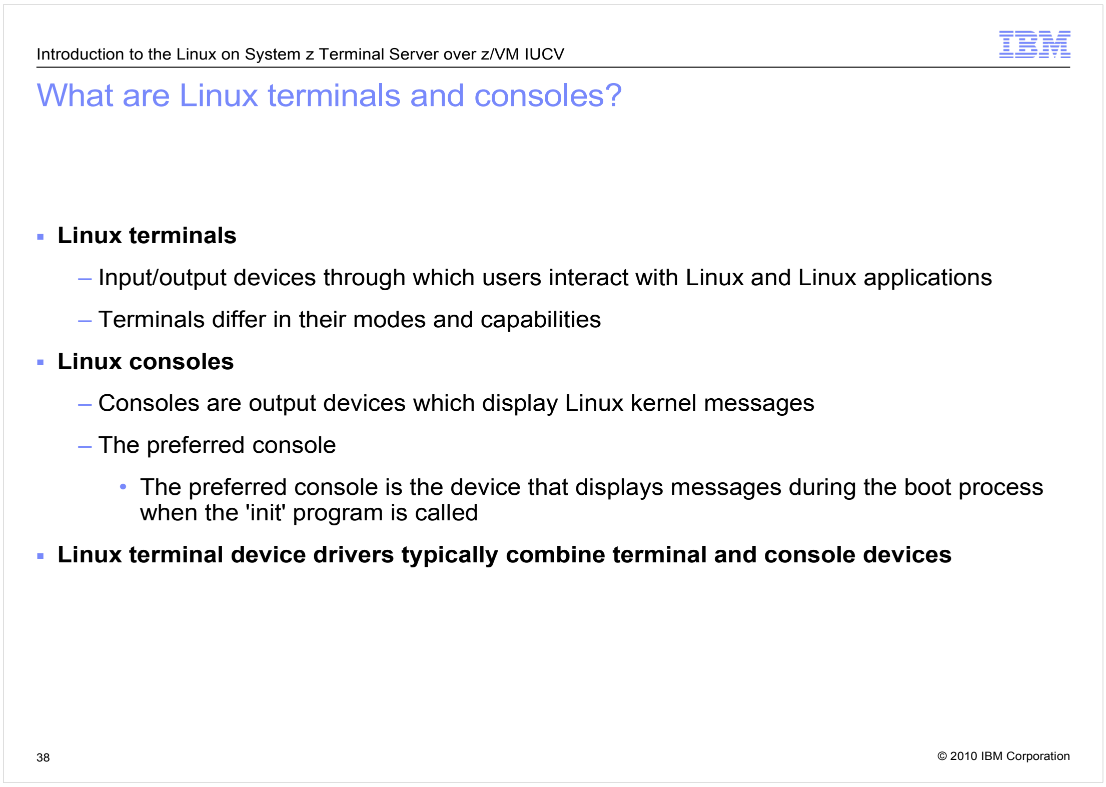
Putting it all together

Target systems

Linux instances on the same z/VM to which IUCV terminal connections are established.

Terminal server

A terminal server is a Linux instance that provides access to terminal devices on other Linux instances, called target systems. The terminal server and all target systems run as guest operating systems of the same z/VM instance. Terminal server and target systems are connected through the z/VM Inter-User Communication Vehicle (IUCV). From the terminal server, administrators can access terminal devices on target systems without requiring direct TCP/IP connections to the target systems.



Introduction to the Linux on System z Terminal Server over z/VM IUCV

What are Linux terminals and consoles?

- **Linux terminals**
  - Input/output devices through which users interact with Linux and Linux applications
  - Terminals differ in their modes and capabilities
- **Linux consoles**
  - Consoles are output devices which display Linux kernel messages
  - The preferred console
    - The preferred console is the device that displays messages during the boot process when the 'init' program is called
- **Linux terminal device drivers typically combine terminal and console devices**

38

Terminal modes:

- Line-mode terminals: 3215
- Block-mode terminals: 3270
- „Full-screen mode“ terminals: allow advanced interactive capabilities

Terminal capabilities:

- Specify terminal initialization sequences, screen positioning, colors, ...

Terminal emulation:

- Today, separate physical terminals are rarities.
- Programs like xterm, kconsole, etc. emulate real terminals.
- The TERM environment variable specifies the set of terminal capabilities. The terminal capabilities are stored in „terminfo“ databases.