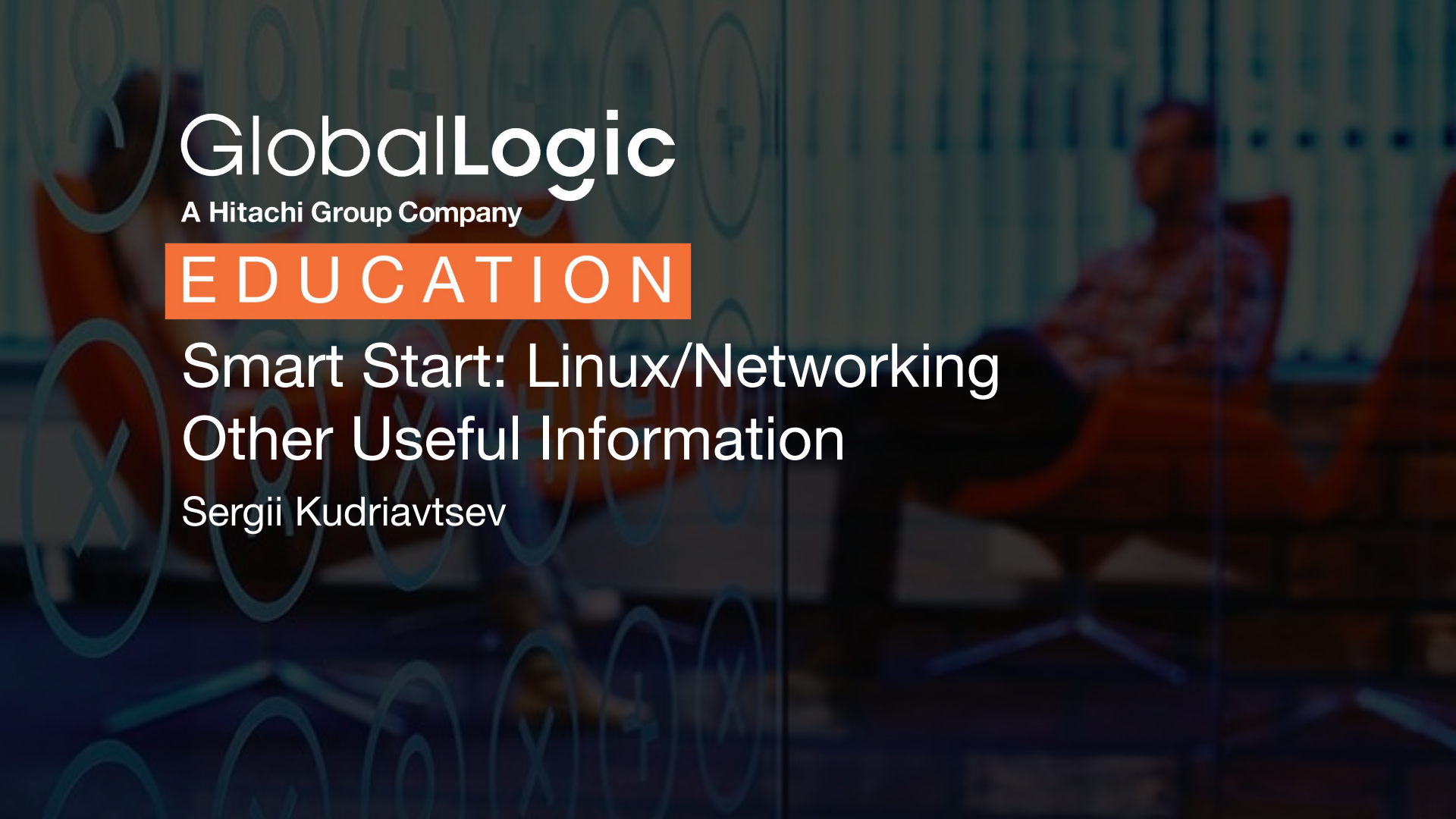


Smart Start: Linux/Networking Other Useful Information

Sergii Kudriavtsev

Agenda

1. hexdump
2. lsof
3. ulimit
4. ps
5. top
6. syslog, dmesg, logs monitoring
7. wget, curl
8. crond
9. archives

hexdump
ASCII, decimal, hexadecimal, octal dump

- $ hexdump *binary*
- $ hexdump -n 304 -C *binary*
- $ hexdump -n 304 -C blk00000.datXXX coreutil

```
00000000  f9 be b4 d9 1d 01 00 00  01 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 3b a3 ed fd  |............;...|
00000030  7a 7b 12 b2 7a c7 2c 3e  67 76 8f 61 7f c8 1b c3  |z{..z.,>gv.a....|
00000040  88 8a 51 32 3a 9f b8 aa  4b 1e 5e 4a 29 ab 5f 49  |..Q2:...K.^J)._I|
00000050  ff ff 00 1d 1d ac 2b 7c  01 01 00 00 00 01 00 00  |......+|........|
00000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 ff ff  |................|
00000080  ff ff 4d 04 ff ff 00 1d  01 04 45 54 68 65 20 54  |..M.......EThe T|
00000090  69 6d 65 73 20 30 33 2f  4a 61 6e 2f 32 30 30 39  |imes 03/Jan/2009|
000000a0  20 43 68 61 6e 63 65 6c  6c 6f 72 20 6f 6e 20 62  | Chancellor on b|
000000b0  72 69 6e 6b 20 6f 66 20  73 65 63 6f 6e 64 20 62  |rink of second b|
000000c0  61 69 6c 6f 75 74 20 66  6f 72 20 62 61 6e 6b 73  |ailout for banks|
000000d0  ff ff ff ff 01 00 f2 05  2a 01 00 00 00 43 41 04  |........*....CA.|
000000e0  67 8a fd b0 fe 55 48 27  19 67 f1 a6 71 30 b7 10  |g....UH'.g..q0..|
000000f0  5c d6 a8 28 e0 39 09 a6  79 62 e0 ea 1f 61 de b6  |\..(.9..yb...a..|
00000100  49 f6 bc 3f 4c ef 38 c4  f3 55 04 e5 1e c1 12 de  |I..?L.8..U......|
00000110  5c 38 4d f7 ba 0b 8d 57  8a 4c 70 2b 6b f1 1d 5f  |\8M....W.Lp+k.._|
00000120  ac 00 00 00 00 f9 be b4  d9 d7 00 00 00 01 00 00  |................|
00000130
```

lsof
list open files

- lsof

List open files

- List of open files by processes
  - `$ lsof`
  - `$ lsof -p PID`
- List of Internet and network files
  - Syntax
    - `$ lsof -i[46][protocol][@hostname|hostaddr][:service|port]`
  - Examples
    - `$ lsof -i`
    - `$ lsof -i4`
    - `$ lsof -i TCP:3333`

- List of open IPv4 sockets
  - Run server listening on port #3333, connect client to it and exchange data
    - Server

      ```
      $ nc -l -s 127.0.0.1 -p 3333
      ```

    - Client

      ```
      $ nc 127.0.0.1 3333
      ```

  - Get PIDs of running `nc` processes:
    - `$ ps -ef | grep 'nc ' | grep -v grep`

```
ironman     9827 19391  0 15:44 pts/27   00:00:00 nc -l -s 127.0.0.1 -p 3333
ironman     9830 25619  0 15:45 pts/30   00:00:00 nc localhost 3333
```

  - List open files of type `IPv4` for `nc` processes
    - `$ p1=9827`
    - `$ p2=9830`
    - `$ lsof -i4 -a -p $p1 -p $p2`

```
COMMAND  PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
nc      9827 ironman    4u   IPv4 215468      0t0  TCP localhost: 3333->localhost:51158
(ESTABLISHED)
nc      9830 ironman    3u   IPv4 215469      0t0  TCP localhost:51158->localhost: 3333
(ESTABLISHED)
```

ulimit
get and set user limits

- Limits
  - ulimit - resources control for shell session
    - `$ ulimit -a # show all limits`
    - `$ ulimit -u # the maximum number of processes available to a single user`
  - Linux
    - /etc/security/limits.conf
      - `$ man limits.conf`
    - sysctl - current kernel settings
      - `$ sysctl -a`
      - `$ sysctl kernel.threads-max`
      - `$ sysctl kernel.threads-max=12000 #under root`
    - /proc/sys - current kernel settings
      - `$ cat /proc/sys/kernel/threads-max`
    - `$ man 5 proc`
  - quota - disk usage and limits

ps
report a snapshot of the current processes

- ps - report a snapshot of the current processes.
  - `$ ps ax # Print all processes for all users`
  - `$ ps # Print Processes for the current user`
  - `$ ps af# print process tree`
  - `$ ps --help all`
  - `$ ps -C minicom -o pid=`

top
display Linux processes

- $ top # Display Linux processes in realtime
- $ top -u username # Display processes of the specified user
- $ top -p PID # Display process PID in real time

syslog
system logger

- System Logs
  - Log dir
    - /var/log
    - /var/log/messages #view under root
    - /var/log/auth.log
    - /var/log/dmesg
    - /var/log/cron
    - /var/log/kern.log
    - /var/log/Xorg.0.log
    - /var/log/lastlog #$ last
    - /var/log/httpd/ or /var/log/apache2/
    - /var/log/[wub]tmp #user access
  - Facility
    - kern, user, mail, daemon, auth, syslog, lpr, news, cron, authpriv(sec), ftp

- System Logs
  - Severity level
    - 0 - Emergency
    - 1 - Alert
    - 2 - Critical
    - 3 - Error
    - 4 - Warning
    - 5 - Notice
    - 6 - Informational
    - 7 - Debug
  - Network protocols
    - UDP - 514 port
    - TCP - 6514 port

- System Logs
  - syslog daemon
    - syslog-ng or rsyslog
    - Installation (under root)
      - `$ apt-get update`
      - `$ apt-get install syslog-ng`
        OR
      - `$ apt-get install rsyslog`
    - start, stop status
      - `$ service syslog-ng status`
        OR
      - `$ service rsyslog status`
    - rsyslog
      - `/etc/rsyslog.conf`
    - logrotate
      - `/etc/logrotate.conf`
      - `/etc/logrotate.d/*`

- System Logs
  - Manual logging utility
    - `$ logger -t MY_DAEMON Vhahah`
  - dmesg
    - Severity level:
      - `emerg, alert, crit, err, warn, notice, info, debug`
    - `Facility:`
      - `Kern, User, Mail, Daemon, Auth, Syslog, Lpr, news`
    - `$ dmesg | tail`
    - `$ dmesg --color -T`
    - `/var/log/dmesg`

- System Logs
  - Logs coloring
    - ccze
      - `$ tail -f /var/log/messages | ccze -A`
  - Logs monitoring
    - logcheck
      - `$ ls -l /etc/logcheck/ /etc/cron*/*logcheck*`
      - `REPORTLEVEL: server, workstation`
    - logwatch
      - `$ ls -l /etc/logwatch/`

wget, curl
The non-interactive network downloader, URL transfering

- wget, curl
  - `$ wget URL`
  - `$ curl URL`

crond
daemon to execute scheduled commands

- crond daemon
  - https://help.ubuntu.com/community/CronHowto #Ubuntu How to
  - `$ service crond status`
  - `$ service cron status`

- 29.3. crontab file
  - Viewing
    - `$ crontab -l`
    - `$ crontab -l -u username`
  - Editing
    - `EDITOR` env variable
    - `$ crontab -e`
    - `$ crontab -e -u username`
  - Example

```
# mins  hours   day     month   weekday command
*/5     *       17,27   *       *       /bin/ls arg1 arg2
```

    - weekday:
      - 0, 7 - Sunday
      - 1 - Monday, 2 - Tuesday, …, 6 - Saturday

archives

- Archives
  - tar
    - packing
      - new archive
        - `$ tar cvf archive.tar file1.txt file2.txt directory1 directory2`
        - `$ tar cvf archive.tar directory`
      - existing archive
        - `$ tar rvf archive.tar new_file`
    - unpacking
      - `$ tar xvf archive.tar`
      - `$ tar -C directory -xvf archive.tar`
    - View
      - `$ tar tvf archive.tar`
    - Extract group of files
      - `$ tar -xvf archive.tar --wildcards '*.jpg'`
    - Compare
      - `$ tar -dvf file.tar -C /path/to/dir`
    - Check archive size
      - `$ cat archive.tar | wc -c #in bytes`

- Archives
  - gzip, gunzip, zcat, zless, zgrep
    - the most default Linux compress utility
    - compressing
      - `$ gzip file.txt # file.txt.gz will be created`
      - `$ gzip archive.tar # file archive.tar.gz will be created`
      - *`$ gzip -k file.txt #do not dell file.txt`*
      - *`$ gzip -r * #will compress files in the main directory as well as all subdirectories`*
      - *`$ gzip -v file.txt #DEBUG INFO`*
      - `$ gzip -c file1.txt > ar.gz; gzip -c file2.txt >> ar.gz; gunzip -c ar.gz`
      - `$ cat test1.txt test2.txt | gzip > foo; gunzip foo`

- Archives
  - gzip, gunzip, zcat, zless, zgrep
    - de-compressing
      - `$ gunzip file.txt.gz # file.txt will be created`
      - `$ gunzip archive.tar.gz # archive.tar will be created`
      - `$ zcat file.txt.gz #unzip archive on the fly and print to stdout`
      - `$ zless file.txt.gz #unzip archive on the fly and pass to "less" pages`
      - `$ zgrep pattern file.txt.gz #unzip archive on the fly and filter by pattern`

- Archives
  - gzip, gunzip, zcat, zless, zgrep
    - Compressing level
    - **-1, --fast**

-2, -3, -4, -5, -6, -7, -8

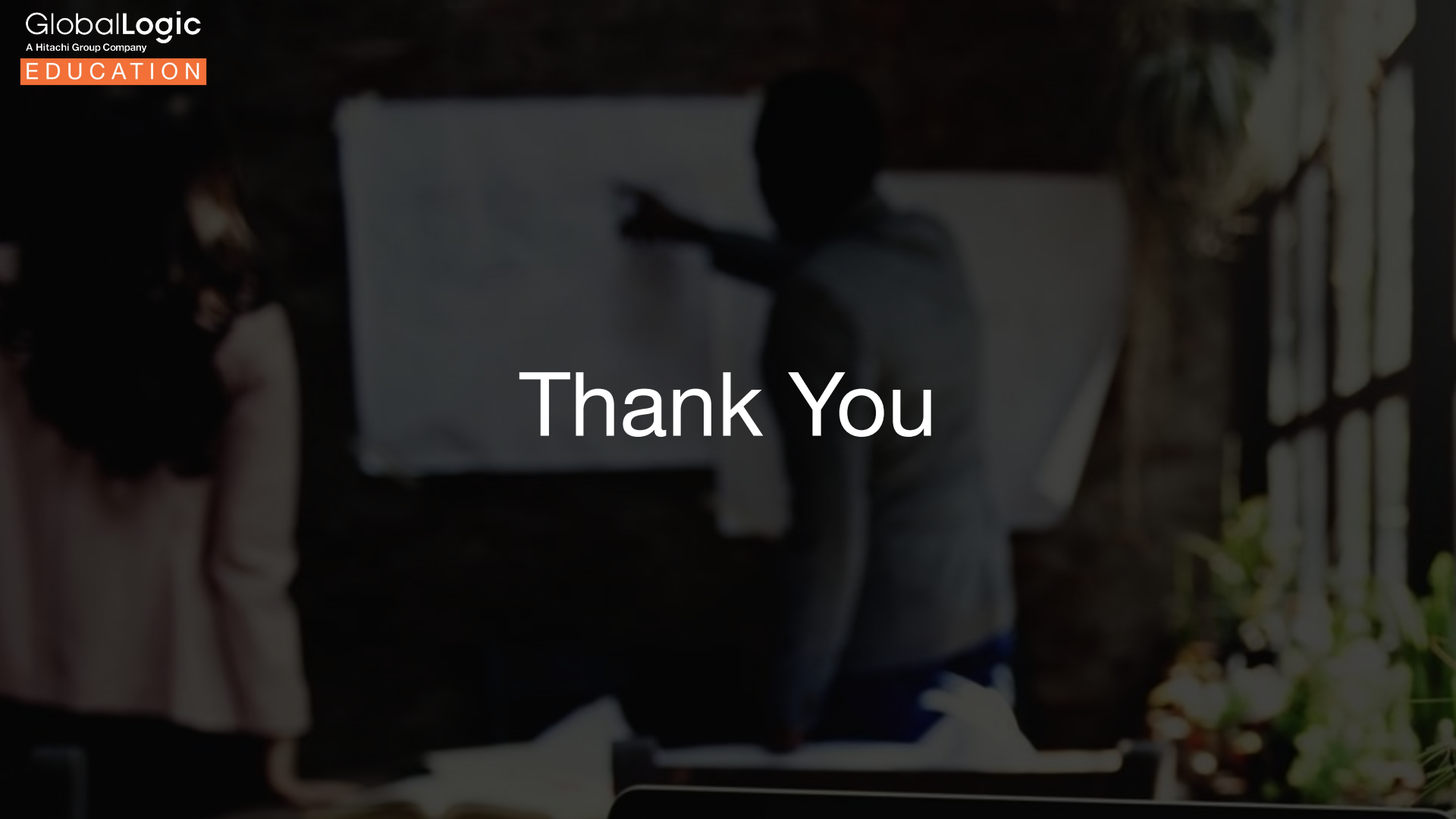
-9, --best These options change the compression level used, with the **-1** option being the fastest, with less compression, and the **-9** option being the slowest, with optimal compression. The default compression level is 6.

- Archives
  - Zip
    - $ sudo apt-get install zip
      - Compress
        - $ zip files.zip file1.txt file2.txt file3.txt
      - Decompress
        - $ unzip *files.zip*
      - Remove file
        - $ zip -d files.zip file3.txt
      - Add new files
        - $ zip -u files.zip file3.txt file4.txt
      - Encrypt
        - $ zip -e file.zip *file1.txt file2.txt file3.txt*
      - Compressing leve
        - (-0, … -9)

- Archives
  - bzip2, bunzip2, bzcat, bzless, bzgrep
    - one more default Linux compress utility
    - a block-sorting file compressor
    - similar to gzip, usually have better compression ratio
      - Compress
        - `$ bzip2 file1.txt file2.txt file3.txt #file1.txt.bz2, file2.txt.bz2 and file3.txt.bz2`
      - Decompress
        - `$ bzip2 -d file.txt.bz2`
      - List compression information
        - `$ bzip2 -v file.txt`
      - Compression level
        - `$ man bzip2`
  - xz, unxz, xzcat, xzless, xzgrep
    - Compress or decompress .xz and .lzma files
    - similar to gzip and bzip2, have better compression ratio, but consumes a lot of CPU and RAM

- Archives
  - tar with gzip and bzip2 (and xz)
    - packing
      - Single command
        - `$ tar czvf archive.tar.gz file1.txt file2.txt directory1 directory2`
        - `$ tar czvf archive.tar.gz directory`
        - `$ tar cjvf archive.tar.bz2 file1.txt file2.txt directory1 directory2`
        - `$ tar cjvf archive.tar.bz2 directory`
        - `$ tar cJvf archive.tar.xz file1.txt file2.txt directory1 directory2`
        - `$ tar cJvf archive.tar.xz directory`
      - Separate commands
        - `$ tar cvf archive.tar file1.txt file2.txt directory1 directory2 && \`
        - `  gzip archive.tar`
        - `$ tar cvf archive.tar directory && gzip archive.tar`
        - `$ tar cvf archive.tar file1.txt file2.txt directory1 directory2 && \`
        - `  bzip2 archive.tar`
        - `$ tar cvf archive.tar directory && bzip2 archive.tar`

- Archives
  - tar with gzip and bzip2 (and xz)
    - unpacking
      - Single command
        - `$ tar xvf archive.tar.gz`
        - `$ tar xvf archive.tar.bz2`
      - Separate commands
        - `$ gunzip archive.tar.gz && tar xvf archive.tar`
        - `$ bunzip2 archive.tar.bz2 && tar xvf archive.tar`
    - view
      - Single command
        - `$ tar tvf archive.tar.gz`
        - `$ tar tvf archive.tar.bz2`
      - Separate commands
        - `$ gunzip archive.tar.gz && tar tvf archive.tar`
        - `$ bunzip2 archive.tar.bz2 && tar tvf archive.tar`


GlobalLogic
A Hitachi Group Company
EDUCATION
Thank You