

C2110 UNIX and programming

13th Lesson

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

National Centre for Biomolecular Research, Faculty of Science
Masaryk University, Kotlářská 2, CZ-61137 Brno



INVESTMENTS IN EDUCATION DEVELOPMENT

CZ.1.07/2.2.00/15.0233

Contents

➤ Data compression

- Lossless versus lossy compression
- Archives
- Archive types, creation, extraction

➤ Source code compilation

- Archive extraction
- Configuration
- Compilation
- Installation

➤ Commands

- gzip, bunzip, bzip2, bunzip2, zip, unzip, tar

Compression

- Lossless
- Lossy

Compression

Compression is process that reduces data (file) size. It is done by searching for **redundant** or **unimportant** information in data, these are then saved more efficiently. According to algorithm used compression may be devided into groups:

- **Lossy compression** – **unimportant information is lost** unreversibly, this is often tolerated by graphical or voice data compression.
- **Lossless compression** – **no information is lost**, compressed data may recover to original state, compression rate is much lower.

Compressed data recovery is **decompression**.

Compression rate denotes compression quality. It is ration of original data size (in bytes) and compressed data size.

Lossy compression

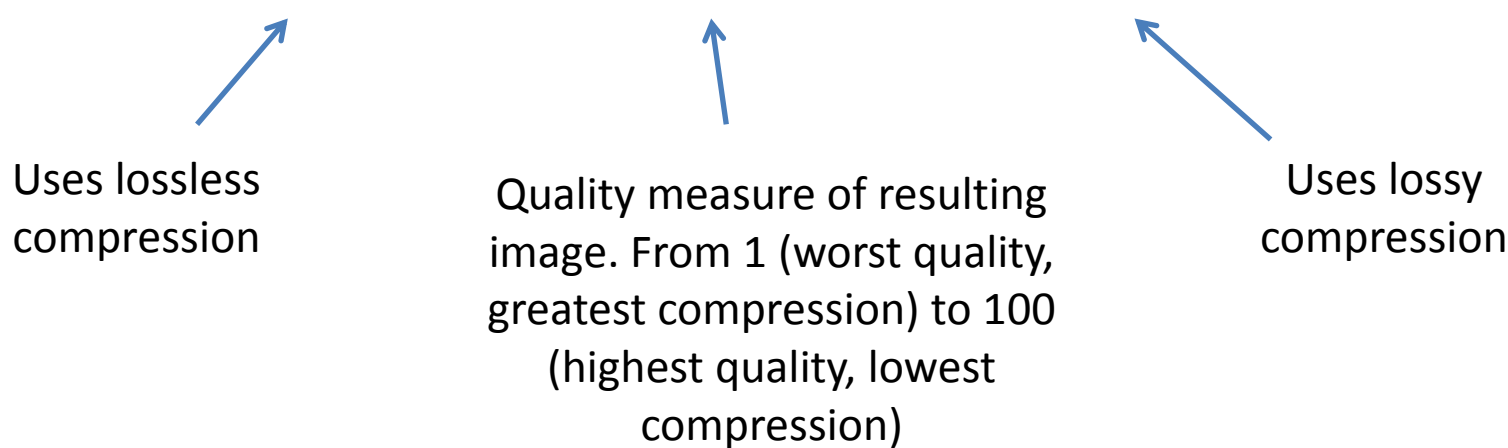
Applications for **lossy compression** and decompression:

- **mplayer**
- **mencoder**
- **convert** (Image Magick)
- and more ...

Conversion of image in PNG format (Portable Network Graphics) to JPEG (Joint Photographic Experts Group):

```
$ convert input.png -quality number output.jpeg
```

Uses lossless
compression



Quality measure of resulting
image. From 1 (worst quality,
greatest compression) to 100
(highest quality, lowest
compression)

Uses lossy
compression

Exercise

1. Copy **test.png** from file **/home/kulhanek/Data/Komprese** to your home directory.
2. What is size of image in bytes?
3. Do **lossy compression** of file to format **jpeg**. Use quality **10**, **50** and **90**. Save outputs to under different names.
4. Compare quality visually (command display).
5. What is **compression rate** for quality 10 and 90?

Lossless compression

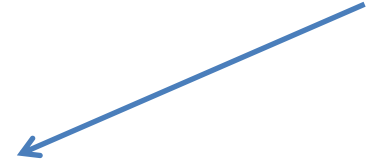
Applications for **lossless** compression and decompression:

- **gzip/gunzip**
- **bzip2/bunzip2**
- **zip/unzip**
- and more ...

Text file **compression**:


```
$ gzip file.txt
```

Output file will be named **file.txt.gz**



```
$ bzip2 file.txt
```

Output file will be named **file.txt.bz2**



Compressed data **decompression**:

```
$ gunzip file.txt.gz
```

```
$ bunzip2 file.txt.bz2
```

Compression and decompression may be done in such a way, that result is written to **standard output** (original file remains unchanged), for example:

```
$ bunzip2 --stdout file.txt.bz2 | wc
```

Exercise

1. Copy text file **bu6_f.log** from directory **/home/kulhanek/Data/Komprese** to your home directory.
2. What is file size in bytes?
3. Do **lossless compression** of file using commands **gzip** and **bzip2**.
4. Which one has higher **compression rate**?
5. Which one does compression **faster**?

Archives

- **Types**
- **Archives creation and extraction**

Archives - tar

In computing, **tar (derived from tape archive)** is both a **file format** (in the form of a type of archive bitstream) and the name of a **program used to handle such files**. The format was created in the early days of Unix and standardized by POSIX standard.

Initially developed to write data to tape backup devices, tar is now commonly used to **collect many files into one larger file** for distribution or archiving, while **preserving file system information** such as user and group permissions, dates, and directory structures.

www.wikipedia.org

Archive extraction:

```
$ tar xvf archive.tar
```

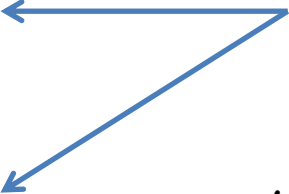
Archive creation:

```
$ tar cvf archive.tar directory/
```

```
$ cd directory
```

```
$ tar cvf /path/to/archive.tar *
```

If archive file name contains extension .gz or .bz2 then archive is automatically decompressed or compressed.



Exercise

1. Find out meaning of option **cvf** of command **tar**?
2. Find out meaning of option **xvf** of command **tar**?
3. Create archive from files files saved in directory: **/home/kulhanek/Data/Archive**
4. What is size of archive file?
5. Do compression of archive. What is **compress rate**?
6. Extract archive to directory **/scratch/your_login/archive**

Source code compilation

- **Application Armagetron**
- **Archive extraction**
- **Configuration**
- **Compilation**
- **Installation**

Armagetron

<http://armagetronad.org/>

Procedure:

- 1) Download source code
- 2) Extract archive
- 3) Read install instructions (README, INSTALL, doc/README, doc/INSTALL)
- 4) Configuration
- 5) Compilation
- 6) Installation

Holy Trinity

```
$ ./configure  
$ make  
$ make install
```

Armagetron, procedure I

Do all steps in scratch.

1) Extract archive:

```
$ tar xvf armagetronad-0.2.8.3.2.src.tar.gz
```

2) Create install directory, i.e. where will be programm installed (necessary if you are not root)

```
$ mkdir armagetronad
```

```
$ pwd
```

```
/scratch/kulhanek/game/armagetronad
```

3) Change current workdirectory to extracted archive data:

```
$ cd armagetronad-0.2.8.3.2
```

Path where will installation be saved.

4) Configuration for compilation and installation:

```
$ ./configure --prefix=/scratch/kulhanek/game/armagetronad \  
--disable-etc --disable-uninstall
```



In this stage, several libraries or applications maybe missing. These may be either installed by similar approach, or more appropriate and faster is to ask administrator to install them. For compilation, development versions of packages has to be installed.

E.g.: # apt-get install libxml2-dev

Armagetron, procedure II

5) Compilation

```
$ make
```

6) Instalation

```
$ make install
```

Path where program is installed.

7) Running program

```
$ cd /scratch/kulhanek/game/armagetronad  
$ bin/armagetronad
```

