Advanced Programming Techniques in Health Care

Health Care IT
WS 2018

mytar
4. and 5. Assignment
(Deadline: 2019-01-29 10:00)

```
USAGE: mytar [--store|--restore|--help] dest src {src}
  --store:   packs all src into [dest]
  --restore: unpacks all files stored in [dest] into the
             current directory
  --help:    this screen
```

You should write a simple C resp. C# archiving program: `mytar`.
`mytar` takes files from the command line and stores them into one big datafile. Also, the files stored in this archive may be unpacked into the current directory.
Only files of the current directory need to be used for input. You do not worry about keeping file permissions and ownerships.
`mytar` creates the destination file and appends every file to store to the big destination file. It keeps track of all stored filenames and -sizes and stores this information as a header before every file data, so it can restore all files by sequentially parsing this big file.
It first reads the actual header, extracts the filename and size, retrieves the following `size` bytes and stores it into `filename`

You have to write 2 variants of the program one in C and one in C# , which are interchangeable.

The following functions may be useful when solving the problem with C:

- `fopen` and `fclose` to open and close a file.
- `fgetc` and `fputc` to read and write a single bytes from/to a `FILE*`.
- `fseek` to position the filepointer in a file, so we could use this to find out the file size, too. Just position the filepointer at `SEEK_END` and read the position with `ftell`.
- Store the file header (filename and -size) as strings, so this information can be easily extracted by using `fscanf` and `fgets`.

big datafile