Parser of MARC21 in ISO 2709

PA193 - Team G (Michal Merta)

December 8, 2014

1 MARC21

MARC21 (MAchine-Readable Cataloging) is standardized format used for structured storage of bibliographic meta data. It's widely used in library catalogs all over the planet. Input of the parser are bibliographic record ISO 2709. Parser converts each record to human readable form. This form is some kind of good practice, there's no standard for it.

2 ISO2709

ISO 2709 is standard for bibliographic description. This format is used for storage of MARC21 format. It can be also used for other versions of MARC. This format has predefined structure. It's plain text format, although unprintable constants `0x1d`, `0x1e`, `0x1f` in order to separate parts of bibliographic records.

3 Parser description

Parser is written in pure C with *std99* dialect. Parser reads records in ISO 2709 from file given on input. File is read chunk by chunk, until constant `0x1d` is found or maximum length of record is reached. Everything from start position of file is stored into dynamically allocated buffer. If end of record is found in the middle of chunk, file is rewind to that position so next record can be read from begin.

Several checks are then performed on the buffer in order to check it's validity. Minimal length of record is check at first. Most of validity testing is done by checking of presence of introduced constants, that has to appear on special indexes in buffer. These indexes are read from predefined positions in buffer. In this part is necessary to convert bytes from parts of buffer into decimal numbers. In valid record there have be numbers on these positions, therefore are these parts tested via regular expressions before attempt to convert them is done. Also every index read from buffer is compared against the total length of record so reading behind buffer is prevented.

Than is buffer part by part converted to human readable for and printed to standard output. If any violation is found, process is stopped and parser exits. In case of successful conversion parser reads another record from file and process is repeated.

4 Summarization

Selected format is not so to parse due to it's inner index structure. Main part of parser consists of moving in buffer read from file and converting the record according to decoded indexes. Working with regular expressions in C for first time was probably the most interesting part. Including handling of inputs, parser consists of 400 lines of code. It's functionality was successfully tested against more than 500 000 publicly available samples of bibliographic meta data.