NAME

zipalign — PKZIP for LLMs

SYNOPSIS

zipalign [FLAG...] *ZIP FILE...*

DESCRIPTION

zipalign adds aligned uncompressed files to a PKZIP archive.

This tool is designed to concatenate gigabytes of LLM weights to an executable. This command goes 10x faster than ‘zip -j0‘. Unlike zip you are not required to use the .com file extension for it to work. But most importantly, this tool has a flag that lets you insert zip files that are aligned on a specific boundary. The result is things like GPUs that have specific memory alignment requirements will now be able to perform math directly on the zip file’s mmap()’d weights.

This tool always operates in an append-only manner. Unlike the InfoZIP *zip*(1) command, **zipalign** does not reflow existing assets to shave away space. For example, if **zipalign** is used on an existing PKZIP archive to replace an existing asset, then the bytes for the old revision of the asset will still be there, along with any alignment gaps that currently exist in the file between assets.

The same concept also applies to the central directory listing that’s stored at the end of the file. When changes are made, the old central directory is left behind as junk data. Therefore it’s important, when adding multiple files to an archive at once, that the files all be passed in arguments at once, rather than calling this command multiple times.

OPTIONS

The following options are available:

`-h` Show help.

`-v` Operate in verbose mode.

`-N` Run in nondeterministic mode. This will cause the date/time of inserted assets to reflect the file modified time.

`-a` *INT*
Byte alignment for inserted zip assets. This must be a two power. It defaults to 65536 since that ensures your asset will be page-aligned on all conceivable platforms, both now and in the future.

`-j` Strip directory components. The filename of each input filepath will be used as the zip asset name. This is otherwise known as the basename. An error will be raised if the same zip asset name ends up being specified multiple times.

`-0` Store zip assets without compression. This is the default. This option must be chosen when adding weights to a llamafile, otherwise it won’t be possible to map them into memory. Using `-0` goes orders of a magnitude faster than using `-6` compression.

`-6` Store zip assets with sweet spot compression. Any value between `-0` and `-9` is accepted as choices for compression level. Using `-6` will oftentimes go 10x faster than `-9` and only has a marginal increase of size. Note uncompression speeds are unaffected.

`-9` Store zip assets with the maximum compression. This takes a very long time to compress. Uncompression will go just as fast. This might be a good idea when publishing archives that’ll be widely consumed via the Internet for a long time.

SEE ALSO

unzip(1), *llamafile*(1)

AUTHORS

Justine Alexandra Roberts Tunney <jtunney@mozilla.com>