xargs Build commands from input

CONCEPTS

xargs turns stdin into arguments: cmd a b c, not cmd fed through its stdin. It packs as many as fit per run.

Producer find, grep, git ls-files
↓ -print0 / lines
xargs split into items
↓ batch -n / -s
Command runs once per batch
↓ -P parallel
Exit status 0, 123, 124..127

Default splitting is on blanks and newlines, and quotes and backslashes are interpreted. Names with spaces break unless you use -0 or -d.

Without -r, GNU xargs runs the command once even on empty input (rm with no args, for example).

BASICS

`<producer>` **| xargs** `<cmd>`
Append input items to a command

xargs `<cmd>` **<** `<file>`
Read items from stdin redirect

xargs -a `<file>` `<cmd>`
Read items from a file

xargs -n `<n>` `<cmd>`
At most n items per run

xargs -L `<n>` `<cmd>`
At most n input lines per run

SPLITTING INPUT

xargs -0 `<cmd>`
Items are NUL-separated

xargs -d '\n' `<cmd>`
Items split on newlines only

find `<dir>` **-print0 | xargs -0** `<cmd>`
Pipe file names safely

SPLITTING INPUT (cont.)

grep -rlZ `<re>` **. | xargs -0** `<cmd>`
Pipe grep matches safely

git ls-files -z | xargs -0 `<cmd>`
Run on tracked files

PLACING ARGUMENTS

xargs -I {} `<cmd>` **{}**
Substitute each item at {}

xargs -I {} sh -c '`<script>` **"$1"' _ {}**
Run a small script per item

-I implies one line per run and overrides -n. Leave it out when batching is faster.

PARALLEL AND CONTROL

xargs -P `<n>` `<cmd>`
Up to n runs at once

xargs -P $(nproc) -n 1 `<cmd>`
One job per CPU core

xargs -r `<cmd>`
Do nothing if input is empty

xargs -t `<cmd>`
Print each command to stderr

With -P, outputs of parallel runs can interleave. Write to per-item files or lock if order matters.

COMMON USES

find . -name '*.log' -print0 | xargs -0 rm
Delete found files

find . -type f -print0 | xargs -0 grep -l `<re>`
Grep many files

cat `<urls>` **| xargs -n 1 -P 8 curl -sO**
Download in parallel