

NAME

capng_change_id – change the credentials retaining capabilities

SYNOPSIS

```
#include <cap-ng.h>
```

```
int capng_change_id(int uid, int gid, capng_flags_t flag);
```

DESCRIPTION

This function will change uid and gid to the ones given while retaining the capabilities previously specified in `capng_update`. It is also possible to specify -1 for either the uid or gid in which case the function will not change the uid or gid and leave it "as is". This is useful if you just want the flag options to be applied (assuming the option doesn't require more privileges that you currently have).

It is not necessary and perhaps better if `capng_apply` has not been called prior to this function so that all necessary privileges are still intact. The caller may be required to have `CAP_SETPCAP` capability still active before calling this function or capabilities cannot be changed. Do not specify `CAP_SETPCAP` in the capability sets unless you intended to have it after the uid change. The library will silently add and remove it if it can. But if its specified, it will leave it alone.

This function also takes a flag parameter that helps to tailor the exact actions performed by the function to secure the environment. The option may be or'ed together. The legal values are:

CAPNG_NO_FLAG

Simply change uid and retain specified capabilities and that's all.

CAPNG_DROP_SUPP_GRP

After changing the primary group ID, remove all supplementary groups. If no new primary group ID is supplied (`gid == -1`), this flag has no effect.

CAPNG_INIT_SUPP_GRP

After changing id, initialize any supplemental groups that may come with the new account. If given with **CAPNG_DROP_SUPP_GRP** it will have no effect.

CAPNG_APPLY_STAGED_GROUPS

Available in libcap-ng 0.9.3 and later. Apply additional groups previously staged with `capng_stage_additional_groups(3)`. If used by itself, the staged list is applied exactly. If combined with **CAPNG_INIT_SUPP_GRP** the account's natural supplemental groups are initialized first and the staged gids are then merged in without duplicates. If combined with **CAPNG_DROP_SUPP_GRP** the call fails. If this flag is set and no staged gids are present, the call fails. Staged gids are intended to be used once and are always cleared before `capng_change_id` returns, even when the flag is omitted or the call fails.

CAPNG_CLEAR_BOUNDING

Clear the bounding set regardless to the internal representation already setup prior to changing the uid/gid.

CAPNG_APPLY_BOUNDING

Available in libcap-ng 0.9.3 and later. Apply the internally prepared bounding set during `capng_change_id`. This only acts on bounding-set state explicitly prepared with `capng_clear(3)`, `capng_fill(3)`, or `capng_update(3)` using **CAPNG_BOUNDING_SET**. If no prepared bounding-set state exists, this flag is a no-op. If combined with **CAPNG_CLEAR_BOUNDING** the call fails.

CAPNG_CLEAR_AMBIENT

Clear ambient capabilities regardless of the internal representation already setup prior to changing the uid/gid.

RETURN VALUE

This returns 0 on success and a negative number on failure.

- 1 means capng has not been initted properly
- 2 means a failure requesting to keep capabilities across the uid change
- 3 means that applying the intermediate capabilities failed
- 4 means changing gid failed
- 5 means initializing supplemental groups failed
- 6 means changing the uid failed
- 7 means dropping the ability to retain caps across a uid change failed
- 8 means clearing the bounding set failed
- 9 means dropping CAP_SETPCAP or ambient capabilities failed
- 10 means resolving the target account for supplemental groups failed
- 11 means dropping supplemental groups failed
- 12 means CAPNG_DROP_SUPP_GRP and CAPNG_APPLY_STAGED_GROUPS were both set
- 13 means CAPNG_APPLY_STAGED_GROUPS was requested with no staged gids
- 14 means applying staged additional groups failed
- 15 means looking up a group account in additional groups failed
- 16 means merging initialized and staged additional groups failed
- 17 means CAPNG_APPLY_BOUNDING and CAPNG_CLEAR_BOUNDING were both set

Applications that need to support libcap-ng releases older than 0.9.3 should detect the availability of **CAPNG_APPLY_BOUNDING** and **CAPNG_APPLY_STAGED_SUPP_GRP** and **capng_stage_additional_groups(3)** before using this flow. Callers that need the prepared bounding-set behavior must request it explicitly with **CAPNG_APPLY_BOUNDING** so older callers continue to see the historical **capng_change_id** behavior.

Note: the only safe action to do upon failure of this function is to probably exit. This is because you are likely in a situation with partial permissions and not what you intended.

SEE ALSO

capng_update(3), **capng_apply(3)**, **capng_stage_additional_groups(3)**, **prctl(2)**, **capabilities(7)**

AUTHOR

Steve Grubb