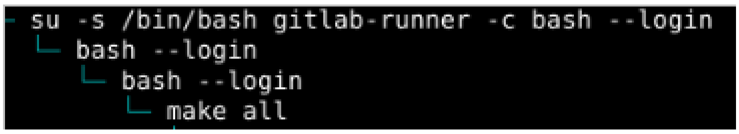
Shell executor

Known issue with job cancellation. Shell executor is using `bash –login user` and for every `script:` section another shell (bash) is spawned. In result proces tree looks like:



```
- su -s /bin/bash gitlab-runner -c bash --login
  └─ bash --login
     └─ bash --login
        └─ make all
```

Default bash behavior is to ignore signals when waiting for child.

```
If bash is waiting for a command to complete and receives a signal for which a trap has been set,
the trap will not be executed until the command completes. When bash is waiting for an asynchronous
command via the wait builtin, the reception of a signal for which a trap has been set will cause
the wait builtin to return immediately with an exit status greater than 128, immediately after which
the trap is executed.
```

The same issue described here:

- Canceling jobs in shell executor doesnt work

Some references:

- Forward SIGTERM to child in Bash
- Gracefully halt cancelled jobs

Solution:

- Using bash wrapper script plus replace current shell with new process using exec

.gitlab-ci.yml

```
build:
  stage: build
  script:
    - exec tools/ci/launch_job.sh make all
    - exec tools/ci/launch_job.sh make deploy
```

tools/ci/launch_job.sh

```
#!/bin/bash
#_term() {
#    echo "$0 caught SIGTERM signal!"
#    trap - SIGTERM
#    kill -TERM $child $$
#}

launch_job() {
    PARENT=$1
    shift
    "$@" &
    CHILD=$!
    trap "trap - SIGTERM; kill -TERM $CHILD $$" SIGTERM

    while sleep 1; do
    if [ ! -e /proc/$PARENT ]; then
        echo "Parent PID $PARENT disappears, terminating child $CHILD"
        kill -TERM $CHILD
        exit
    fi

    if [ ! -e /proc/$CHILD ]; then
        wait $CHILD
        trap - SIGTERM
        exit $?
    fi
    done
}

#trap _term SIGTERM
#launch_job $$ "$@" &
#echo "$0 Waiting for $! to finish"
#wait $!

launch_job $PPID "$@"
```

From:
https://niziak.spox.org/wiki/ - **niziak.spox.org**

Permanent link:
https://niziak.spox.org/wiki/git:gitlab:runner:shell

Last update: **2022/01/24 13:25**