

Praca odbywa się w kontenerze

1. Ściągamy źródła do naszego workspace/

```
$ git clone https://github.com/crosstool-ng/crosstool-ng  
$ cd crosstool-ng/
```
2. Konfigurujemy środowisko dla toolchaina

```
$ ./bootstrap  
$ ./configure --prefix=${PWD}
```
3. Kompilacja źródeł

```
$ make  
$ make install
```
4. Dodajemy źródła do PATH

```
$ export PATH="${PWD}/bin:${PATH}"
```

Środowisko jest gotowe - teraz pozostaje skonfigurować i skompilować docelowy toolchain dla naszego targetu - Raspberry Pi. Interesuje nas architektura *64 bitowa (aarch64)*

Zadanie: Za pomocą narzędzia `ct-ng` znajdź konfigurację (sample), którą użyjemy jako podstawę naszego toolchaina.

Wywołanie:

```
$ ct-ng
```

Pokaże dostępne komendy dla tego narzędzia.

Przykład: Sprawdzenie obecnej konfiguracji można zrobić przez wywołanie:

```
$ ct-ng show-config
```

Jak używać podpowiedzi - po zaznaczeniu myszką linijki powinna być widoczna. Jeśli nie - trzeba skopiować i wkleić do innego edytora.

Hint1:

Hint2:

Hint3:

5. Sprawdzamy konfigurację
\$ ct-ng show-aarch64-rpi4-linux-gnu
6. Wprowadzamy i zapisujemy
\$ ct-ng aarch64-rpi4-linux-gnu
7. Zmiana docelowego folderu, gdzie będą binarki toolchaina (uwaga: komenda w jednej linii)
\$ sed -i
/^CT_PREFIX_DIR=/s/.*\/CT_PREFIX_DIR="\${CT_PREFIX:-\${PWD}}\/x-tools}\/\${CT_HOST:+HOST-\${CT_HOST}}\/}\${CT_TARGET}"/' .config
8. Wynik komendy:
\$ grep CT_PREFIX_DIR .config
CT_PREFIX_DIR="\${CT_PREFIX:-\${PWD}}\/x-tools}\/\${CT_HOST:+HOST-\${CT_HOST}}\/}\${CT_TARGET}"

--- Pauza ---

Zadanie: Za pomocą

\$ ct-ng menuconfig

Zmień następujące parametry:

Wersja kernela - **5.1.21**

Wersja glibc - **2.37**

Zaznacz aby został zbudowany program do debugowania **'strace'**

Zapisz nową konfigurację

Sprawdź czy została nadpisana (przez \$ ct-ng show-config)

Efekt końcowy powinien wyglądać tak:

```
pokyuser@8cad83cdbfbb:/workdir/crosstool-ng$ ct-ng show-config
[1...] aarch64-rpi4-linux-gnu
  Languages      : C,C++
  OS             : linux-5.1.21
  Binutils       : binutils-2.40
  Compiler       : gcc-13.2.0
  C library      : glibc-2.37
  Debug tools    : gdb-13.2 strace-6.5
  Companion libs : expat-2.5.0 gettext-0.21 gmp-6.2.1 isl-0.26
libiconv-1.16 mpc-1.3.1 mpfr-4.2.1 ncurses-6.4 zlib-1.2.13
zstd-1.5.5
  Companion tools :
```

9. Budowanie - ostatni krok (zajmie ok 20 minut - do odpalenia w przerwie)
\$ ct-ng build