



## Laborator 08

Scopul laboratorului este crearea environment-ului pentru lucru cu OpenCL pe procesoare GPU (CPU dacă nu aveți GPU).

Pentru a putea rula codul creat de voi pe checker va trebui să putem să compilăm pentru Linux.

Din Windows:

**Nu vom folosi WSL.** VSCode va merge direct pe Windows.

1. Avem nevoie de să rulăm [GCC din Visual Studio Code](#).
2. În ghid la pasul 3 urmează instalarea [MSYS2](#) Mingw. Asigurați-vă că urmați ghidul de pe site-ul [MSYS2](#).
3. După ce terminați instalare reveniți la [site-ul anterior](#) și continuați de la pasul 4. Va trebui să adăugați MSYS2 bin în path,
4. În acest moment ar trebui să puteți rula gcc dintr-un terminal din Windows, inclusiv powershell.
5. Instalați extensia de VSCode pentru [OpenCL](#).
6. Deschideți View -> Command Pallet
7. Instalați suport OpenCL pentru GPU-ul, sau CPU-ul vostru

GPU/CPU	Download
<b>NVIDIA</b>	<a href="#">Driver GPU</a>
<b>Intel</b>	<a href="#">SDK Intel OpenCL</a>
<b>AMD</b>	<a href="#">Driver GPU/CPU</a>
<b>Apple</b>	Gata instalat

8. Compilați și rulați programul dat ca exemplu din schelet. Din terminal, în funcție de sistem folosiți unul din exemplele de mai jos:

```
gcc test.c \
-L "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.4\lib\x64" \
-I "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.4\include" \
-IOpenCL

gcc test.c \
-L "C:\Program Files (x86)\IntelSWTools\system_studio_2020\OpenCL\sdk\lib\x64" \
-I "C:\Program Files (x86)\IntelSWTools\system_studio_2020\OpenCL\sdk\include" \
-IOpenCL
```