

```
Activities Terminal lis 11 20:49
student
Trash

student@ubuntu224-06: ~
.global _start
_start:
    mov x0, #1
    ldr x1, =msg
    mov x2, #13
    mov x8, #64
    svc #0

    mov x0, #0
    mov x8, #93
    svc #0

student@ubuntu224-06:~$ aarch64-linux-gnu-as -o hello_world.o hello_world.s
student@ubuntu224-06:~$ ls
Desktop  Downloads  hello_world.s  Pictures  snap  tsclient
Documents  hello_world.o  Music  Public  Templates  Videos
student@ubuntu224-06:~$ aarch64-linux-gnu-ld -o hello_world hello_world.o
student@ubuntu224-06:~$ file hello_world
hello_world: ELF 64-bit LSB executable, ARM aarch64, version 1 (SYSV), statically linked, not stripped
student@ubuntu224-06:~$ qemu-aarch64
qemu: no user program specified
student@ubuntu224-06:~$ qemu-aarch64 ./hello_world
Hello, World
student@ubuntu224-06:~$
```

Zad.2

Moim zdaniem, z punktu widzenia użytkownika, x86-64 jest prostsze, ponieważ taki kod działa bezpośrednio na komputerach z tą architekturą, co upraszcza cały proces. Dodatkowo x86-64 używa 16 rejestrów, co jest łatwiejsze do opanowania. W moim odczuciu syscall jest prostsze w użyciu od svc #0. ARM64 wymaga używania np. QEMU do uruchomienia kodu na komputerze z architekturą x86, co sprawiło u mnie małe komplikacje i rozwiązanie zadania zajęło mi więcej czasu niż poprzednie.

Liliana Jaśkiewicz, gr. K1, numer albumu: 95098