**zadanie 1**

**podpunkt a)**

-o: zapis wyników do pliku

-e: filtrowanie wywołań systemowych

-p: śledzenie procesu na podstawie PID

-c: zliczanie statystyk wywołań systemowych

**podpunkt b)**

**użyte komendy:** strace ./hello

Kluczowym wywołaniem, które odpowiada za wyświetlenie napisu "Hello, world" na standardowym wyjściu jest write(1, “Hello world \n”, 13).

Funkcja printf w C jest używana do formatowania i wyświetlania tekstu. Po wywołaniu printf, przekłada się to na wywołanie systemowe write, które odpowiada za zapis danych na standardowe wyjście.

**podpunkt c)**

**użyte komendy:** strace -e trace=open,write ./hello

, gdzie opcja -e trace=open,close filtruje wywołania systemowe związane z otwieraniem i wypisywaniem

**otrzymany tekst:** write(1, “Hello world \n”, 13)

, gdzie 1 oznacza deskryptor standardowego wyjścia stdout

**popdunkt d)**

Program wyświetli napis w formacie: Hello, <username>, gdzie <username> to wartość zmiennej środowiskowej LOGNAME

Wywołanie getenv będzie widoczne w śledzeniu wywołań systemowych jako odczyt zmiennej środowiskowej.

Strace śledzi wywołania systemowe, takie jak operacje na plikach I procesy, a ltrace śledzi wywołania funkcji bibliotecznych, takich jak te z standardowej biblioteki C.

Aby jednocześnie śledzić wywołania funkcji systemowych i funkcji bibliotecznych, można skorzystać z wywołania:

strace z opcją -e trace=all:

**podpunkt e)**

strace -e trace=file -o plik.txt sh hello.sh

grep "open" strace\_output.txt | grep -E "\.bashrc|\.profile|\.bash\_profile|\.bash\_logout|\.bash\_login|\.inputrc|\.shrc"