





ZAD. 2  
 Wygeneruj plik nekawolny <swap.o>  
 po czym na podstawie wydruku poleceń  
 readelf -t -s dla każdego elementu  
 tabeli symboli podaj:

ELEMENT	NAL. SEKCJI	NUMER	TYP	ROZMIAR	ADRES
swap-p.c	absolute not in any section		local	0	0...0
bufp1	4 (.bss)		local	8	0...0
incr	1 (.text)		local	22	0...0
count.1758	4 (.bss)		local	4	0...8
bufp0	5 (.data)		global	8	0...0
buf	und		global (extern)	0	0...0
initialvalue	3 (.data)		global	4	0...0
swap	1 (.text)		global	77	0...16(17)

Co przechowują sekcje .strtab i .shstrtab

.strtab (string table) - zawiera nazwy  
 zmiennych i funkcji programu

.shstrtab (section header string table) - zawiera  
 nazwy sekcji

ZAD. 3 Po uruchomieniu programu  
 drukuje pewien ciąg znaków i kolejny  
 działając bez zgłaszania błędów.  
 Czym tak się dzieje?

Program wypisze 0x48 (bo char main to  
 1 bajt), czyli wypisze pierwszy bajt  
 funkcji main).

Zakończy działanie bez zgłaszania  
 błędów, ponieważ main jest symbolem  
 globalnym w elf.c natomiast statycznym w  
 nob.c, a p2 jest symbolem statycznym w  
 main.c ale globalnym w nob.c co jest  
 zgodne z zasadami linkowania.

po przypisaniu zmiennej main wartości  
 w funkcji p2 nastąpi naruszenie ochrony  
 pamięci.

Jeśli zainicjujemy zmienną main w  
 miejscu deklaracji to program nie  
 skompiluje się, ponieważ niedozwolone  
 są dwa użycia tego samego zmiennego.



ZAD. 4 Które wiersze w kodzie z  
rod. 2 będą wymagały dodania wpisu  
do tablicy relokacji?

83 to wiersze 8 i 9 (symbol lokalny),  
13 (referencja symboli zewnętrznych i  
lokalnego), 14 (referencja symboli globalnego i lokalnego),  
15 (referencja symboli globalnego i lokalnego),  
16 (referencja symboli lokalnego).

W sekcji .data.rel.local będzie zmienna  
buf (to można sprawdzić za pomocą  
readelf -t -S) i ona ma mieć podsta-  
wiony adres zmiennej zewnętrznej, buf  
e to może zrobić tylko linker w  
procesie relokacji.

ZAD. 5

UNDEF - symbol, do którego <sup>nie</sup> odwołujemy  
się jest zdefiniowany w innym pliku.  
Możemy zauważyć, że w tabeli symboli  
nie mamy zdefiniowanego typu ani  
rozmiaru buf, ponieważ ma on  
signature extern

COMMON - symbol niezainicjalizowany z  
definicją w tym samym pliku

Wszystkie obiekty COMMON są zbierane  
przez linker podczas tworzenia pliku  
wykonywalnego i w zależności od  
stawień (lub nie, wyitek stanowi tutaj  
inicjalizacja zerem) inicjalizacji  
zaczynane w sekcji .data lub .bss. Jeśli  
nie ma więcej niż jedna inicjalizacja tego  
samego symbolu to linker poinformuje  
nas o błędzie.