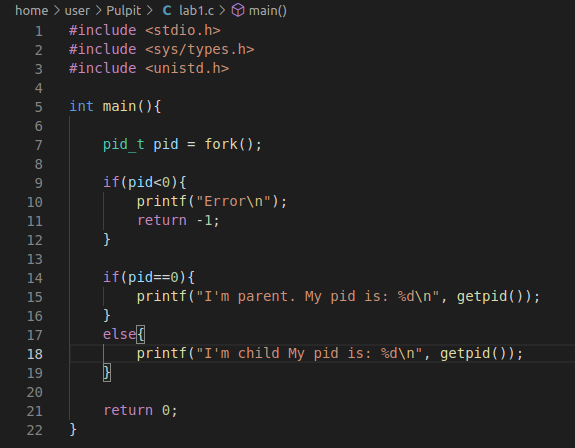
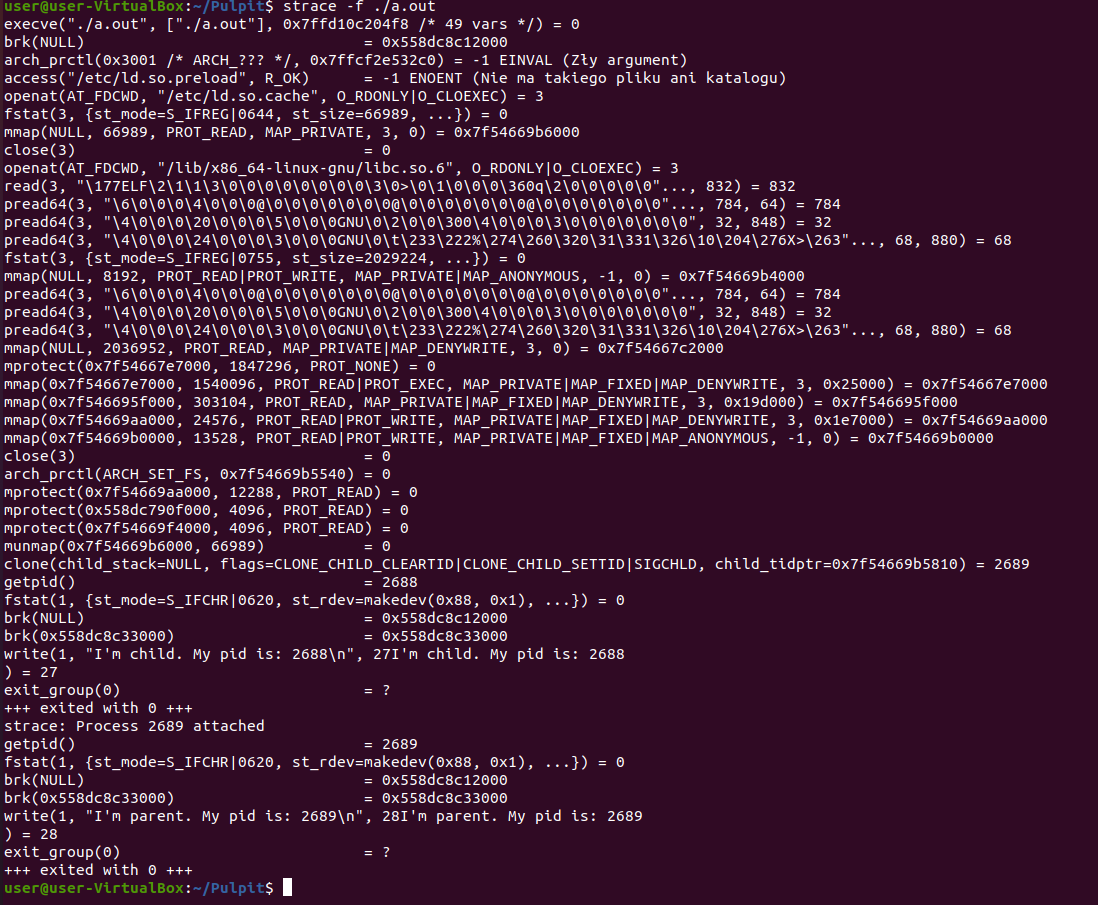
# Zadanie 1

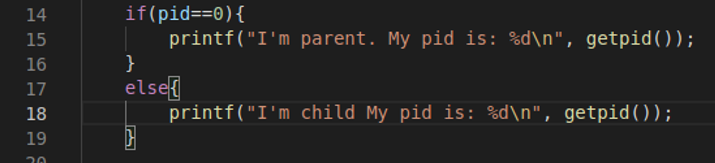
Kod programu:  


# 



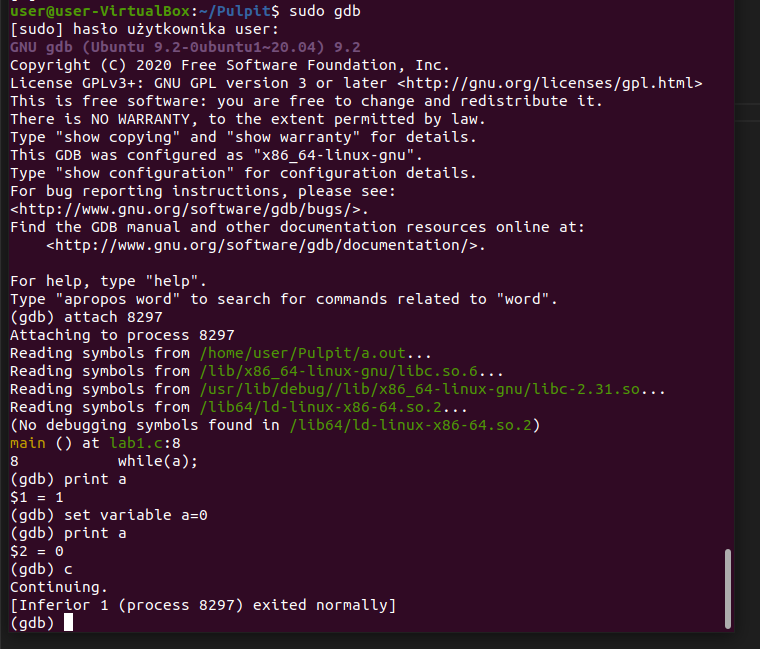
execve() – odpowiada za uruchomienie programu w nowym procesie  
clone() – odpowiada wywołaniu funkcji fork(), czyli za zduplikowanie procesu (linia 7)

  
getpid() – wywołanie funkcji getpid(), która zwraca nam numer PID (linia 15 i 18)  
write() – odpowiada wywołaniu printf() (linia 15 i 18)



# Zadanie 2

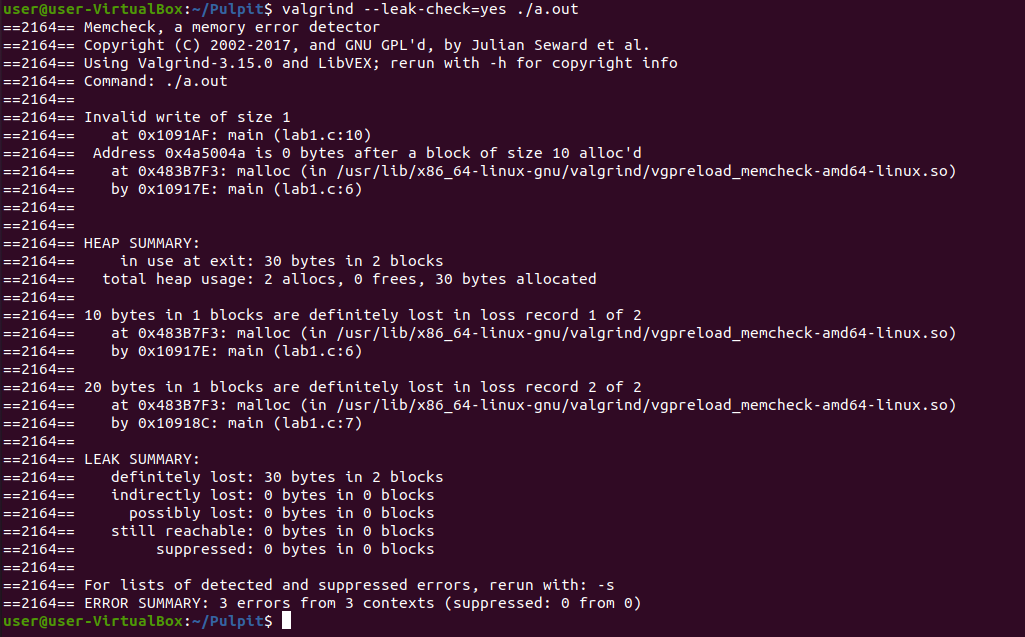
# 



Wstępnie skompilowałem program z przełącznikiem -g. Uruchomiłem program w tle wykorzystując ./a.out& otrzymałem w ten sposób pid procesu. Następnie uruchomiłem gdb z uprawnieniami roota i podpiąłem się do procesu wykorzystując attach pid. Wypisałem zmienna a oraz zmieniłem ją na 0, następnie kontynuowałem wykonywanie programu poleceniem c, przez co program się zakończył.

# Zadanie 3



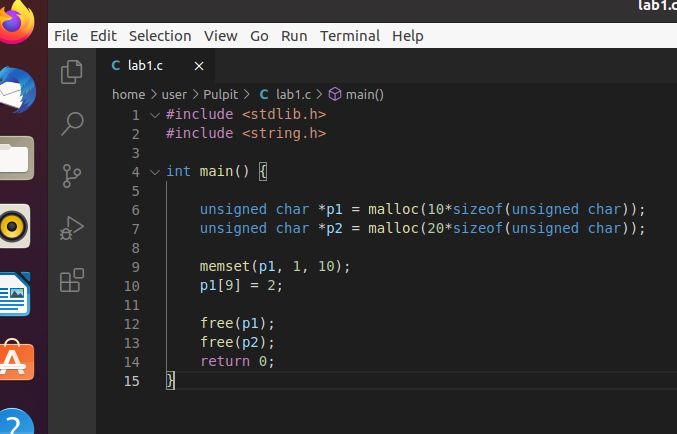


Raport programu Valgrind wykazał, że mamy 3 podatności w naszym programie

Dwie dotyczą braku zwalniania pamięci tablic typu unsigned char powołanych w (liniach 6 i 7), należałoby użyć polecenia free(), gdyby tablice były powoływane do życia w funkcji która jest często wykonywalna, nasza pamięć mogła by się zapełnić.

Trzeci błąd dotyczy próby zapisu do tablicy poza przydzieloną pamięcią (linia 10), nasza tablica zawiera 10 elementów a indeks 10 odnosi się do 11 elementu. Jeżeli chcemy odnieść się do ostatniego elementu należy użyć 9 indeksu.

Kod po poprawkach



Raport Valgrinda po poprawkach: