Laboratorium 2

Przemysław Ziaja
Systemy Operacyjne
Akademi Górniczo-Hutnicza

March 24, 2020

Zadanie 1

Co to sa deskryptory plików?
Jest to identyfikator pliku używany przez system operacyjny (w POSIX typ całkowity(int)). Po otwarciu pliku może być wykorzystywany wielokrotnie przez system operacyjny.

Jakie sa standardowe deskryptory otwierane dla procesów?
0 - STDIN
1 - STDOUT
2 - STDERR

Jakie flagi trzeba ustawić w funkcji open aby otrzymać funkcjonalność funkcji creat?
O_CREAT

W wyniku wykonania polecenia umask otrzymano 0022. Jakie prawa dostepu bedzie miał plik otwarty w nastepujacy sposób:
open(pathname, O_RDWR |O_CREAT, S_IRWXU |S_IRWXG |S_IRWXO)
755 ponieważ komenda chce założyć plik z prawami 777 ale dodajemy do tego XOR maski i wychodzi 755.

Co oznaczaja flagi: O_WRONLY |O_CREAT |O_TRUNC?
O_WRONLY - WRite only
O_CREAT - Utwórz plik jeżeli nie istnieje
O_TRUNC - Jeżeli plik istnieje to zmniejsz jego długość do 0 (takie zapisz jako)

Co oznacza flaga O_APPEND?
O_APPEND - Dopisuje do pliku, tzn. wszystkie operacje pisania beda odbywały sie na końcu pliku

Co oznacza zapis: S_IRUSR |S_IWUSR?
S_IRUSR - Read USeR - prawo do odczytu dla użytkownika
S_IWUSR - Write USeR - prawo do zapisu dla użytkownika

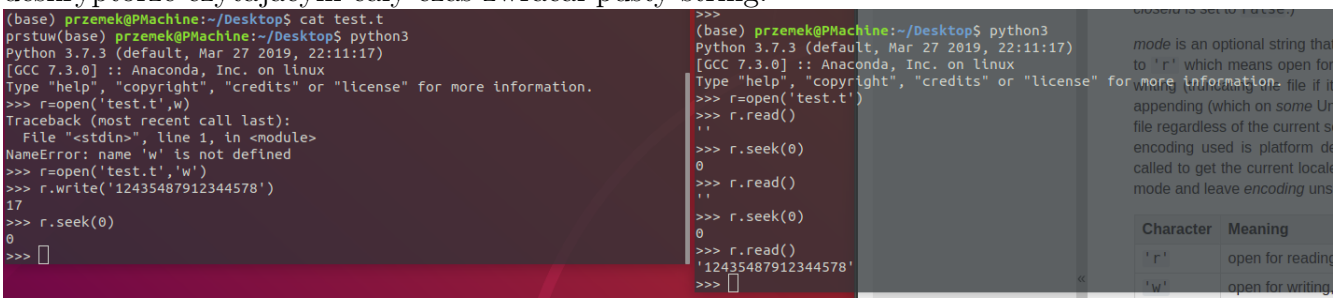
Zadanie 2

Czy w momencie powrotu z funkcji write dane sa już zapisane na urzadzenie wyjściowe?
Nie, system nie musi od razu wrzucić danych na np. dysk, może trzymać je w buforze i zapisywać, a program bedzie kontynuowany.

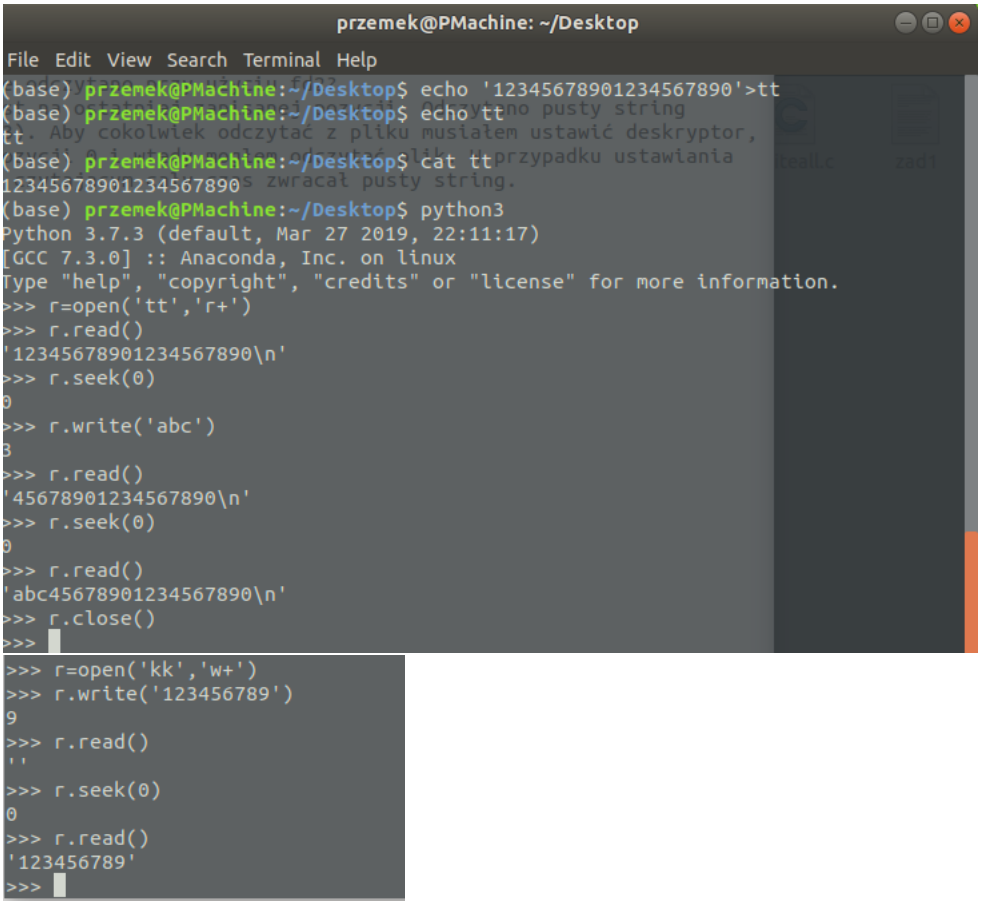
Co robi ta funkcja? Jakiej sytuacji dotyczy wartość EINTR?
Wpisuje dane z o długości nbyte z bufora do pliku otwartego przy pomocy deskryptora fd.
EINTR - kod błedu przechowywany w errno, stosowany do oznaczenia przrwania wywołania funkcji.

Zadanie 3

Dwa deskryptory: fd1 i fd2 użyto do otwarcia pliku podajac te same ścieżke dostepu do pliku. Wskaźnik pliku ustawiony jest na poczatku pliku. Nastepnie korzystajac z deskryptora fd1 wykonano operacje zapisania 100b do pliku. Nastepnie przy użyciu deskryptora fd2 wykonano operacje czytania z pliku. Pytanie: Na jakiej pozycji jest wskaźnik pliku? Jakie dane odczytano przy użyciu fd2?
Wskaznik pliku jest na ostatniej zapisanej pozycji. Odczytano pusty string (dołaczam zrzuty z python3). Aby cokolwiek odczytać z pliku musiałem ustawić deskryptor, który zapisywał dane do pozycji 0 i wtedy mogłem odczytać plik. W przypadku ustawiania pozycji 0 na deskryptorze czytajacym cały czas zwracał pusty string.



Do otwarcia pliku użyto jednego deskryptora fd3. Nastepnie wykonano kolejno operacje pisania 100b i czytania 100b. Na jakiej pozycji jest wskaźnik pliku? Co zostało przeczytane?
Jezeli plik istnial wczesniej to zostanie przeczytana reszta pliku, jezeli tworzymy go dopiero to zostanie przeczytany pusty string.(screeny)



Czy każdorazowe poprzedzenie operacji pisania ustawieniem wskaźnika pliku na końcu pliku za pomoca funkcji lseek daje taki sam rezultat jak otwarcie pliku w trybie z ustawiona flaga O_APPEND? Odpowiedź uzasadnij.
If the O_APPEND file status flag is set on the open file description, then a write(2) always moves the file offset to the end of the file, regardless of the use of lseek().
manual

O_APPEND guarantees that every write to the file will be at the end of the file. lseek guarantees that the file pointer is positioned to the current EOF of the file. It does not guarantee that subsequent writes will be at the end of the file.
jim mcnamara
`https://www.unix.com/programming/203467-what-main-difference-between-difference-between-append.html`

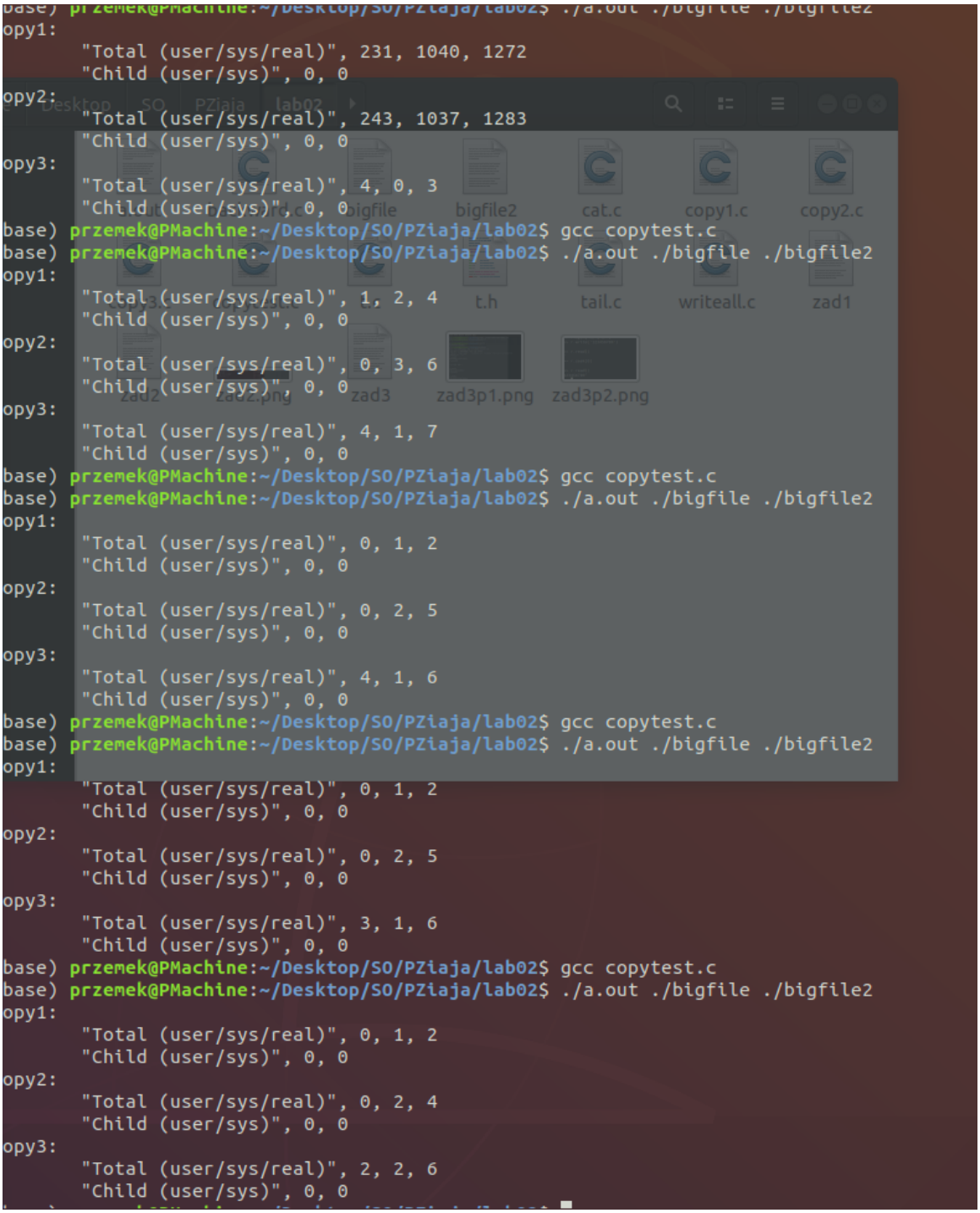
Czyli:
O_APPEND - koniec pliku
lseek - koniec pliku podczas edytowania

Jak wyglada wywołanie funkcji lseek które:
ustawia wskaźnik na zadanej pozycji?
znajduje koniec pliku?
zwraca bieżaca pozycje wskaźnika?

lseek(deksryptor, pozycja, 0);
lseek(desktyptor, 0L, 2);
lseek(dekstyptor,0L,1);

Zadanie 4

Jak wiadać na poniższym obrazku różnica w czasie pomiedzy plikami jest znacza, ale spowodowane jest to rozmiarem bufora. Gdy bufor w porównaniu do rozmiaru pliku jest mały różnica jest marginalna. Wtedy czas wykonania operacji jest długi i opłaca sie zainwestować w dodatkowe zabezpieczenia. Podobnie jest w przypadku małych plików lub gdy rozmiar bufora w porównaniu do pliku jest rozmiaru 0.1%. Czas wykonania jest na tyle krótki, że lepiej zainwestować w zabezpieczenia, niż mierzyć sie z potencjalnymi problemami jeżeli zaniecha sie tych kroków.


opy1:
"Total (user/sys/real)", 231, 1040, 1272
"Child (user/sys)", 0, 0
opy2:
"Total (user/sys/real)", 243, 1037, 1283
"Child (user/sys)", 0, 0
opy3:
"Total (user/sys/real)", 4, 0, 3
"Child (user/sys)", 0, 0
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ gcc copytest.c
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ ./a.out ./bigfile ./bigfile2
opy1:
"Total (user/sys/real)", 1, 2, 4
"Child (user/sys)", 0, 0
opy2:
"Total (user/sys/real)", 0, 3, 6
"Child (user/sys)", 0, 0
opy3:
"Total (user/sys/real)", 4, 1, 7
"Child (user/sys)", 0, 0
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ gcc copytest.c
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ ./a.out ./bigfile ./bigfile2
opy1:
"Total (user/sys/real)", 0, 1, 2
"Child (user/sys)", 0, 0
opy2:
"Total (user/sys/real)", 0, 2, 5
"Child (user/sys)", 0, 0
opy3:
"Total (user/sys/real)", 4, 1, 6
"Child (user/sys)", 0, 0
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ gcc copytest.c
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ ./a.out ./bigfile ./bigfile2
opy1:
"Total (user/sys/real)", 0, 1, 2
"Child (user/sys)", 0, 0
opy2:
"Total (user/sys/real)", 0, 2, 5
"Child (user/sys)", 0, 0
opy3:
"Total (user/sys/real)", 3, 1, 6
"Child (user/sys)", 0, 0
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ gcc copytest.c
base) przemek@PMachine:~/Desktop/SO/PZiaja/lab02$ ./a.out ./bigfile ./bigfile2
opy1:
"Total (user/sys/real)", 0, 1, 2
"Child (user/sys)", 0, 0
opy2:
"Total (user/sys/real)", 0, 2, 4
"Child (user/sys)", 0, 0
opy3:
"Total (user/sys/real)", 2, 2, 6
"Child (user/sys)", 0, 0
bigfile
bigfile2
cat.c
copy1.c
copy2.c
t.h
tail.c
writeall.c
zad1
zad3
zad3p1.png
zad3p2.png