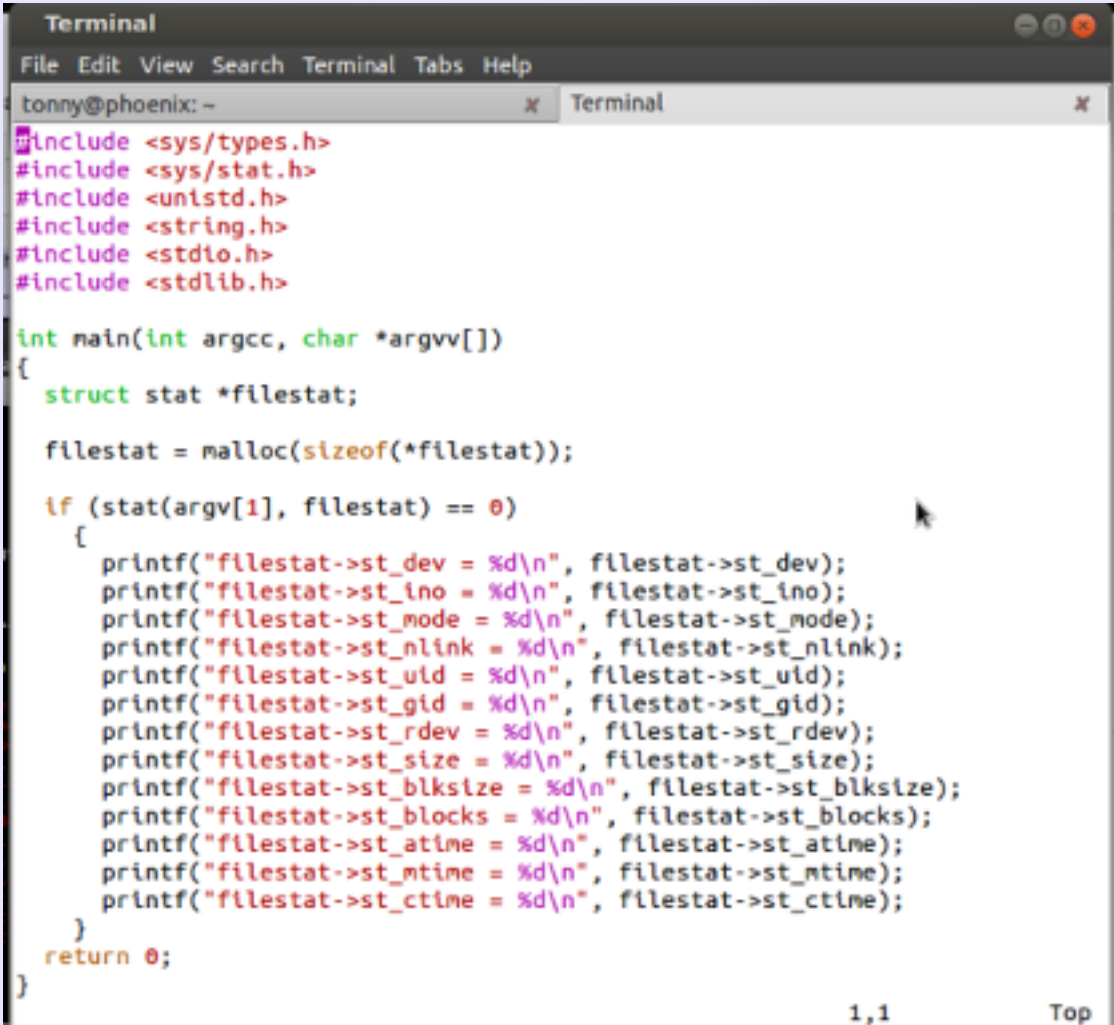


| | | |
|-----------------|-------------------------------------|--|
| NAMA: | | |
| NPM: | | |
| PR2 | | System Calls (lanjutan) + Stat |
| PERIODE: | | |
| ASISTEN: | | TS |
| | | RMP |
| | | RS |
| | | TD |
| | | FI |
| | | KD |
| | | PD |
| | | |
| No | Kemampuan | |
| 1 | Persiapan | |
| | * | Buka Virtual Box dan import berkas ova yang anda export dari kuliah. |
| | * | Masuk ke direktori kerja anda (WORK) dan update repository anda dengan versi terbaru di projects |
| | | # cd WORK |
| | | # svn update |
| | Hint | |
| | | |
| 2 | Kelompok Soal Pertama : Redirection | |
| | | Rizki menemukan sebuah script yang mengandung redirection berikut : |
| | | \$cat script.sh |
| | | #!/bin/bash |
| | | #tonny.adhi@ui.ac.id |
| | | #Contoh dengan Script Redirection |
| | * | grep \$1 << EOF |
| | | Ramot 0211234778 |
| | | Rizki 0226788999 |
| | | Adin 0811234556 |
| | | Tegar 08561243245 |
| | | Win 08590342356 |
| | | EOF |
| | | Ketika dieksekusi , script tersebut mengeluarkan output sebagai berikut : |
| | * | \$/script.sh Ramot |
| | | Ramot 0211234778 |
| | | \$/script.sh Win |
| | | Win 0859032356 |

| | | | |
|---|--|---|--|
| | * | Apa yang terjadi saat perintah di atas dieksekusi? Jelaskan fungsi redirection beserta contohnya. | |
| 3 | Kelompok Soal Kedua : Stat System Call | | |
| | * | Menurut Anda system call apa saja yang dipakai saat kita mengeksekusi perintah "ls -al"? Berikan alasannya ! | |
| | * | Diberikan sebuah program C sederhana yang menggunakan stat system call sebagai berikut : | |
| | |  <pre> Terminal File Edit View Search Terminal Tabs Help tonny@phoenix: ~ #include <sys/types.h> #include <sys/stat.h> #include <unistd.h> #include <string.h> #include <stdio.h> #include <stdlib.h> int main(int argc, char *argv[]) { struct stat *filestat; filestat = malloc(sizeof(*filestat)); if (stat(argv[1], filestat) == 0) { printf("filestat->st_dev = %d\n", filestat->st_dev); printf("filestat->st_ino = %d\n", filestat->st_ino); printf("filestat->st_mode = %d\n", filestat->st_mode); printf("filestat->st_nlink = %d\n", filestat->st_nlink); printf("filestat->st_uid = %d\n", filestat->st_uid); printf("filestat->st_gid = %d\n", filestat->st_gid); printf("filestat->st_rdev = %d\n", filestat->st_rdev); printf("filestat->st_size = %d\n", filestat->st_size); printf("filestat->st_blksize = %d\n", filestat->st_blksize); printf("filestat->st_blocks = %d\n", filestat->st_blocks); printf("filestat->st_atime = %d\n", filestat->st_atime); printf("filestat->st_mtime = %d\n", filestat->st_mtime); printf("filestat->st_ctime = %d\n", filestat->st_ctime); } return 0; } </pre> | |

| | | |
|---|--|--|
| | * | Program tersebut menggunakan sebuah input berupa file yang akan dibaca metadatanya. Berikut merupakan contoh kompilasi eksekusi dari program tersebut |
| | | <pre>#gcc -w -o statcall statcall.c && ./statcall jawaban.txt filestat->st_dev = 2049 filestat->st_ino = 208210 filestat->st_mode = 16893 filestat->st_nlink = 1 filestat->st_uid = 1000 filestat->st_gid = 1000 filestat->st_rdev = 0 filestat->st_size = 3752 filestat->st_blksize = 4096 filestat->st_blocks = 7 filestat->st_atime = 1378780119 filestat->st_mtime = 1378780116 filestat->st_ctime = 1378780116</pre> |
| | * | Dapatkah kalian sebutkan dan jelaskan atribut apa saja yang ditampilkan dari sebuah file yang dibaca metadata-nya oleh syscall tersebut ? Hint, kalian juga dapat mengkompilasi dan menjalankan program sederhana di atas dengan input yang berbeda . |
| 4 | Kelompok Soal Ketiga : File Descriptor | |
| | * | Saat kita ingin melihat file descriptor untuk process id, kadang kita mendapat output, kadang tidak. Sebagai contoh |
| | | <pre>root@fafner:/tmp# ls /proc/7477/fd 0 1 10 11 12 14 15 2 3 4 5 6 7 8 9 root@fafner:/tmp# ls /proc/20/fd</pre> |
| | * | Menurut Anda, sekumpulan angka yang ditampilkan tersebut sebenarnya apa? Coba jelaskan! |
| | * | Kenapa kita tidak mendapatkan output yang sama untuk setiap process ID ? Adakah process ID yang tidak memiliki file descriptor ? Coba jelaskan ! |
| 5 | Kelompok Soal Keempat: /dev/null | |
| | * | Tonny sedang melihat - lihat isi /dev, dan dia tertarik melihat /dev/null. Dia kemudian melakukan perintah berikut: |
| | | <pre>root@fafner: gcc prog.c > /dev/null</pre> |
| | * | Apa yang terjadi saat Tonny mengeksekusi perintah di atas? |
| | * | Bagaimanakah cara kerja /dev/null? Apa saja kegunaannya? Berikan contoh. |

