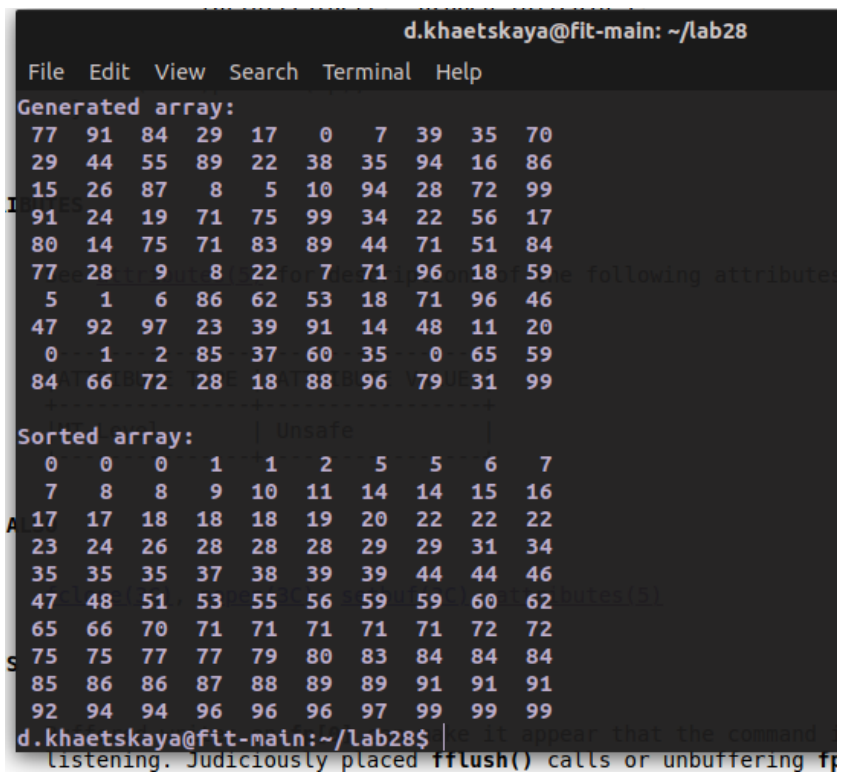
OS Lab 28

Status	approved
checkbox	☑
class	OS
due date	@Apr 14, 2021

Task

28. Генератор случайных чисел

Напишите программу, которая генерирует сортированный список из ста случайных чисел в диапазоне от 0 до 99.
Распечатайте числа по десять в строке.
Используйте `p2open(3)`, чтобы запустить `sort(1)` и `rand(3)` и `srand(3)` для генерации случайных чисел.




```
The rand() function uses a multiplicative congruential ran
dom-number
       generator with period 2^32 that returns successive
 pseudo-random numbers
       in the range of 0 to RAND_MAX (defined in <stdlib.h
>).


       The  srand() function uses the argument seed as a s
eed for a new sequence
       of pseudo-random numbers to be returned by subseque
nt calls to rand(). If
       srand() is then called with the same seed value, th
e sequence of pseudo-
       random numbers will be repeated.  If rand() is call
ed before any calls to
       srand() have been made, the same sequence will be g
enerated as when
       srand() is first called with a seed value of 1.
```

```
The p2open() function forks and execs a shell running the
 command line
       pointed to by cmd. On return, fp[0] points to a FIL
E pointer to write the
       command's standard input and fp[1] points to a FILE
pointer to read from
       the command's  standard output.  In this way the pr
ogram has control over
       the input and output of the command.


       The function returns 0 if successful; otherwise, it
returns -1.


The p2close() function is used to close the file pointers
 that p2open()
       opened.  It waits for the process to terminate and
 returns the process
       status. It returns 0 if successful; otherwise, it r
eturns -1.
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <libgen.h>

int main() {
  srand(time(NULL));
  FILE * fd[2];
  if( p2open("sort", fd) == -1){
    perror("p2open failed");
    return -1;
  }

  int currentNum;

  printf("Generated array:\n");
for(int i = 0; i < 10; i ++){
  for(int j = 0; j < 10; j++){
    currentNum = rand() % 100;
    fprintf(fd[0], "%d\n", currentNum);
    printf("%3d ", currentNum);
  }
  printf("\n");
}
if (pclose(fd[0]) == -1){
  perror("pclose failed");
  return -1;
}

printf("\n");

printf("Sorted array:\n");
for(int i = 0; i < 10; i++){
  for(int j = 0; j < 10; j++){
    currentNum = rand() % 100;
    fscanf(fd[1], "%d", &currentNum);
    printf("%3d ", currentNum);
  }
  printf("\n");
}
```

http://src.illumos.org/source/xref/illumos-gate/usr/src/lib/libc/port/stdio/popen.c?r=462453d2#89

```
  if (pclose(fd[1]) == -1){
    perror("pclose failed");
    return -1;
  }
  return 0;
}
```

```
58
59  typedef struct node {
60          pid_t   pid;
61          int     fd;
62          struct  node    *next;
63  } node_t;
64
```