ITMO University

The Faculty of Software Engineering and Computer Systems

System software basics

Practical task №4

C programming

Variant 1

Done by
a student of P3310 group
Dima Glushkov

Saint-Petersburg
2019

Task 1.

Using syscalls write c programm (c89/c99 standarts) which gonna perform like `cat` system tool. Also implement a few keys of the tool and make it able to work with stdin in case `-` given instead of file name.

Solution:

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <fcntl.h>

void wrong_keys();
void print(int);
void vprint(int, char, char);


int main(int argc, char * argv[])
{
    char eflag=0, tflag=0;
    char * file, key;
    int fd;
    while ((key = getopt(argc, argv, "eEtTuA")) != -1)
        switch (key) {
               case 'e':
               case 'E':
                    eflag = 1;
                    break;
               case 't':
               case 'T':
                    tflag = 1;
                    break;
               case 'A':
                    eflag = 1;
                    tflag = 1;
                    break;
               case 'u':
                    break;
            default:
                wrong_keys();
        }
    file = *(argv + optind);
     if (!strcmp(file, "-") || argc == 0)
          fd = 1;
     else if ((fd = open(file, O_RDONLY)) == -1)
     {
        perror(strerror(errno));
          exit(1);
     }
```

```
     if (eflag || tflag)
          vprint(fd, eflag, tflag);
     else
          print(fd);
     close(fd);
    return 0;
}

void wrong_keys()
{
    fprintf(stderr, "\nError: usage cat [-eEtTuA] [file ...]\n");
    exit(1);
}

void print(int fd)
{
     char sym;
     while(read(fd, &sym, 1) > 0)
          write(1, &sym, 1);
}

void vprint(int fd, char eflag, char tflag)
{
     char sym;
     while(read(fd, &sym, 1) > 0)
     {
          if (eflag && sym == '\n')
               write(1, "$", 1);
          if (tflag && sym == '\t')
          {
               write(1, "^I", 2);
               continue;
          }
          write (1, &sym, 1);
     }
}
```

Task 2.
Write perl programm (c89/c99 standarts) which gonna perform perform like `cat` system tool.
Solution:

```
#!/usr/bin/perl -T

use strict;
use warnings qw(FATAL all);

if (defined $ARGV[0])
{
     my $filename = $ARGV[0];

     if ($filename eq "-")
     {
          while (<>)
          {
               print;
          }

     }
     else
     {
          open(FH, '<', $filename) or die $!;
          while(<FH>)
          {
               print $_;
          }
     }
}
else
{
     die("Error: Specify file");
}
```

Task 3.
Write perl programm (c89/c99 standarts) which gonna perform perform like `xargs` system tool.
Solution:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>

#define BUFFSIZE 80

int main(int argc, char * argv[])
{
```

```
     if (argc > 1)
     {
          char keys[BUFFSIZE] = {0};

          fgets(keys, BUFFSIZE, stdin);

          puts(keys);
          if (keys[strlen(keys)-1] == '\n')
          {
               system(strcat(strcat(argv[1], " "), keys));
          }
          else
          {
               puts("Error: input is to long");
          exit (1);
          }

     }
     else
     {
          puts("Error: no command specified");
          puts("Usage: xxargs [command]");
          exit(1);
     }

}
```