Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації та управління

Лабораторна робота №3

З дисципліни «Сучасні операційні системи»

Студента групи ІС-82

Кучер-Савінський Назарій

Варіант № 12

Київ

2021

Зображення, що містить текст

Автоматично згенерований опис

Лістинг програми

#include <stdio.h>

#include <unistd.h>

#include <fcntl.h>

#include <sys/types.h>

#include <sys/stat.h>

int main(int argc, char \*\* argv) {

  int status;

  char \* cat\_arg\_list[] = {

    "cat",

    argv[1],

    NULL

  };

  char \* grep1\_arg\_list[] = {

    "grep",

    "28/Oct/2016",

    NULL

  };

  char \* cut\_arg\_list[] = {

    "cut",

    "-d",

" ",

    "-f7",

    NULL

  };

  char\* sort1\_arg\_list[] = {

    "sort",

    NULL

    };

  char\* uniq\_arg\_list[] = {

    "uniq",

    "-c",

     NULL

    };

   char\* sort2\_arg\_list[] = {

    "sort",

    "-nr",

    NULL

   };

   char\* head\_arg\_list[] = {

   "head",

   "-10",

   NULL

   };

  char \* awk\_arg\_list[] = {

    "awk",

    "BEGIN { FS=\" \" } { result[$2] = $1 } END { for (x in result) { total += result[x] }  for (x in result) { printf \"%s - %d - %.2f%% \\n\", x, result[x], result[x]\*100/total } }",

    NULL

  };

  char\* sort3\_arg\_list[] = {

   "sort",

   "-nrb",

   "-t ",

   "-k3",

   NULL

   };

  char\* nl\_arg\_list[] = {

   "nl",

   NULL

  };

  int pipes[20];

  pipe(pipes);

  pipe(pipes + 2);

  pipe(pipes + 4);

  pipe(pipes + 6);

  pipe(pipes + 8);

  pipe(pipes + 10);

  pipe(pipes + 12);

  pipe(pipes + 14);

  pipe(pipes + 16);

  pipe(pipes + 18);

  if (fork() == 0) {

    dup2(pipes[1], 1);

    for (int i = 0; i < 20; i++)

    {

       close(pipes[i]);

    }

    execvp( \* cat\_arg\_list, cat\_arg\_list);

  } else {

    if (fork() == 0) {

      dup2(pipes[0], 0);

      dup2(pipes[3], 1);

      for (int i = 0; i < 20; i++)

      {

         close(pipes[i]);

      }

      execvp( \* grep1\_arg\_list, grep1\_arg\_list);

    } else {

      if (fork() == 0) {

        dup2(pipes[2], 0);

        dup2(pipes[5], 1);

        for (int i = 0; i < 20; i++)

        {

           close(pipes[i]);

        }

        execvp( \* cut\_arg\_list, cut\_arg\_list);

      } else {

        if (fork() == 0) {

          dup2(pipes[4], 0);

          dup2(pipes[7], 1);

          for (int i = 0; i < 20; i++)

          {

             close(pipes[i]);

          }

          execvp ( \* sort1\_arg\_list, sort1\_arg\_list);

        } else {

          if (fork() == 0) {

            dup2(pipes[6], 0);

            dup2(pipes[9], 1);

            for (int i = 0; i < 20; i++)

            {

               close(pipes[i]);

            }

            execvp( \* sort1\_arg\_list, sort1\_arg\_list);

          } else {

            if (fork() == 0) {

              dup2(pipes[8], 0);

              dup2(pipes[11], 1);

              for (int i = 0; i < 20; i++)

              {

                 close(pipes[i]);

              }

              execvp( \* uniq\_arg\_list, uniq\_arg\_list);

            } else {

              if (fork() == 0) {

                dup2(pipes[10], 0);

                dup2(pipes[13], 1);

                for (int i = 0; i < 20; i++)

                {

                 close(pipes[i]);

                }

                execvp( \* sort2\_arg\_list, sort2\_arg\_list);

              } else {

                if (fork() == 0) {

                  dup2(pipes[12], 0);

                  dup2(pipes[15], 1);

                  for (int i = 0; i < 20; i++)

                  {

                   close(pipes[i]);

                  }

                  execvp( \* head\_arg\_list, head\_arg\_list);

                } else {

                  if (fork() == 0) {

                    dup2(pipes[14], 0);

                    dup2(pipes[17], 1);

                    for (int i = 0; i < 20; i++)

                    {

                     close(pipes[i]);

                    }

                    execvp( \* awk\_arg\_list, awk\_arg\_list);

                  } else {

                    if (fork() == 0) {

                      dup2(pipes[16], 0);

                      dup2(pipes[19], 1);

                      for (int i = 0; i < 20; i++)

                      {

                       close(pipes[i]);

                      }

                      execvp( \* sort3\_arg\_list, sort3\_arg\_list);

                    } else {

                      if (fork() == 0) {

                        dup2(pipes[18], 0);

                        for (int i = 0; i < 20; i++)

                        {

                         close(pipes[i]);

                        }

                        execvp( \* nl\_arg\_list, nl\_arg\_list);

                      }

                    }

                  }

                }

              }

            }

          }

        }

      }

    }

  }

  for (int i = 0; i < 20; i++)

  {

      close(pipes[i]);

  }

  for (int i = 0; i < 19; i++)

     wait(&status);

}

Результат роботи з huge-access.log:

Зображення, що містить текст

Автоматично згенерований опис