|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБАРАТОРНОЙ РАБОТЫ №5**

по дисциплине: «Операционные системы»

|  |  |  |
| --- | --- | --- |
| Студент |  | Беркович Евгений Сергеевич |
| Группа |  | РК6-55Б |
| Тип задания |  | Конвейерная оценка повторений слов в текстовых файлах |
| Вариант |  | 02 |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_ **Беркович Е. С.**\_ \_

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** \_ **Грошев С. В.** \_\_

*подпись, дата фамилия, и.о.*

*Москва, 2023 г.*

**Техническое задание**

Требуется разработать программу для поиска слов, состоящих из латинских букв, в указанных текстовых файлах. В заданном текстовом файле нужно перечислить все различные слова, состоящие из латинских букв без учета регистра, которые имеют указанную длину. Список слов должен отображаться в алфавитном порядке

Выполнение программы должны обеспечивать 6 параллельных процессов, которые взаимодействуют через 5 программных каналов, образуя конвейер команд. Указанные процессы должны выполняться по программам следующих команд OS UNIX: cat, tr ( -cs), sort, uniq ( -c), sort ( -h) и tail (- l). Стандартный вывод во всех процессах, кроме последнего, выполняющего команду tail, должен быть перенаправлен в программный канал связи со следующим процессом. Стандартный ввод во всех процессах, кроме первого, выполняющего программу cat, должен быть заменен вводом из программного канала связи с предыдущим процессом. Полученный результат должен отображаться через стандартный вывод в процессе, выполняющем программу tail. Все процессы, образующие указанный конвейер команд, должны порождаться основным процессом выполнения программы, который должен ожидать их завершения.

**Приложение №1**

**Содержимое файла popular.c**

|  |
| --- |
| /\* Popular file words \*/  #include "sys/wait.h"  #include "sys/types.h"  #include "unistd.h"  #include "stdlib.h"  int main(int argc, char\* argv[]) {  int fdp1[2];  int fdp2[2];  int fdp3[2];  int fdp4[2];  int fdp5[2];  pipe(fdp1);  pipe(fdp2);  pipe(fdp3);  pipe(fdp4);  pipe(fdp5);  if(argc < 2) {  write(1, "Usage: popular file-name words-number\n", 38);  exit(0);  }    if(fork() == 0) {  close(1); dup(fdp1[1]); close(fdp1[1]); close(fdp1[0]);  execl("/bin/cat", "cat", argv[1], (char\*) 0);  exit(1);  }  if(fork() == 0) {  close(0); dup(fdp1[0]); close(fdp1[0]); close(fdp1[1]);  close(1); dup(fdp2[1]); close(fdp2[1]);  execl("/usr/bin/tr", "tr", "-cs", "A-Za-z", "\n", (char\*) 0);  exit(2);  }  if(fork() == 0) {  close(0); dup(fdp2[0]); close(fdp2[0]); close(fdp2[1]);  close(fdp1[0]); close(fdp1[1]);  close(1); dup(fdp3[1]); close(fdp3[1]); close(fdp3[0]);  execl("/usr/bin/sort", "sort", (char\*) 0);  exit(3);  }  if(fork() == 0) {  close(0); dup(fdp3[0]); close(fdp3[0]); close(fdp3[1]);  close(fdp1[0]); close(fdp1[1]);  close(fdp2[0]); close(fdp2[1]);  close(1); dup(fdp4[1]); close(fdp4[1]);  execl("/usr/bin/tail", "tail", "-n", argv[2], (char\*) 0);  exit(4);  }  if(fork() == 0) {  close(0); dup(fdp4[0]); close(fdp4[0]); close(fdp4[1]);  close(fdp1[0]); close(fdp1[1]);  close(fdp2[0]); close(fdp2[1]);  close(fdp3[0]); close(fdp3[1]);  close(1); dup(fdp5[1]); close(fdp5[1]);  execl("/usr/bin/sort", "sort", (char\*) 0);  exit(5);  }  if(fork() == 0) {  close(0); dup(fdp5[0]); close(fdp5[0]); close(fdp5[1]);  close(fdp1[0]); close(fdp1[1]);  close(fdp2[0]); close(fdp2[1]);  close(fdp3[0]); close(fdp3[1]);  close(fdp4[0]); close(fdp4[1]);  execl("/usr/bin/uniq", "uniq", "-c", "-i", (char\*) 0);  exit(6);  }  close(fdp1[0]); close(fdp1[1]);  close(fdp2[0]); close(fdp2[1]);  close(fdp3[0]); close(fdp3[1]);  close(fdp4[0]); close(fdp4[1]);  close(fdp5[0]); close(fdp5[1]);  while(wait(0) != (-1));  exit(0);  } |

**Содержимое файла text.txt**

This is an example of text that includes multiple repeated Words. Repeated Words can make text quite interesting. Can you identify which words are repeated? I bet you can! You can use the program we discussed to count how often individual words are used. This could be a fun learning exercise. Exercise your programming skills and it will benefit you in future learning endeavours. So, isn't this a good way to learn more about word frequency analysis?

**Приложение №2**

**Результат работы программы:**

