**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 **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ**

**ТЕХНОЛОГИИ» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

**О Т Ч Е Т**

**по лабораторной работе №** 4

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Название:** | Защищенный режим |  |
| **Дисциплина:** | Операционные системы |  |
| Студент | ИУ7-53Б | А.С.Саркисов |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | (Группа) |  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель |  |  |  |  | Н.Ю. Рязанова |

(Подпись, дата) (И.О. Фамилия)

Москва, 2020

Задание №1

Написать программу, запускающую не мене двух новых процессов си- стемным вызовом fork(). В предке вывести собственный идентификатор (функция getpid()), идентификатор группы ( функция getpgrp()) и иден- тификаторы потомков. В процессе-потомке вывести собственный иден- тификатор, идентификатор предка (функция getppid()) и идентификатор группы. Убедиться, что при завершении процесса-предка потомок, кото- рый продолжает выполняться, получает идентификатор предка (PPID), равный 1 или идентификатор процесса-посредника.

Ниже приведён листинг программы для задания №1.

Graphical user interface, text, application, email

Description automatically generated



Graphical user interface, text, application, email

Description automatically generated



Ниже приведён пример работы программы для задания №1.

A picture containing text, plaque, scoreboard

Description automatically generated

Задание №2

Написать программу по схеме первого задания, но в процессе-предке выполнить системный вызов wait(). Убедиться, что в этом случае иден- тификатор процесса потомка на 1 больше идентификатора процесса- предка.

Ниже приведён листинг программы для задания №2. Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Ниже приведён пример работы программы для задания №2.

A picture containing text, plaque

Description automatically generated



Задание №3

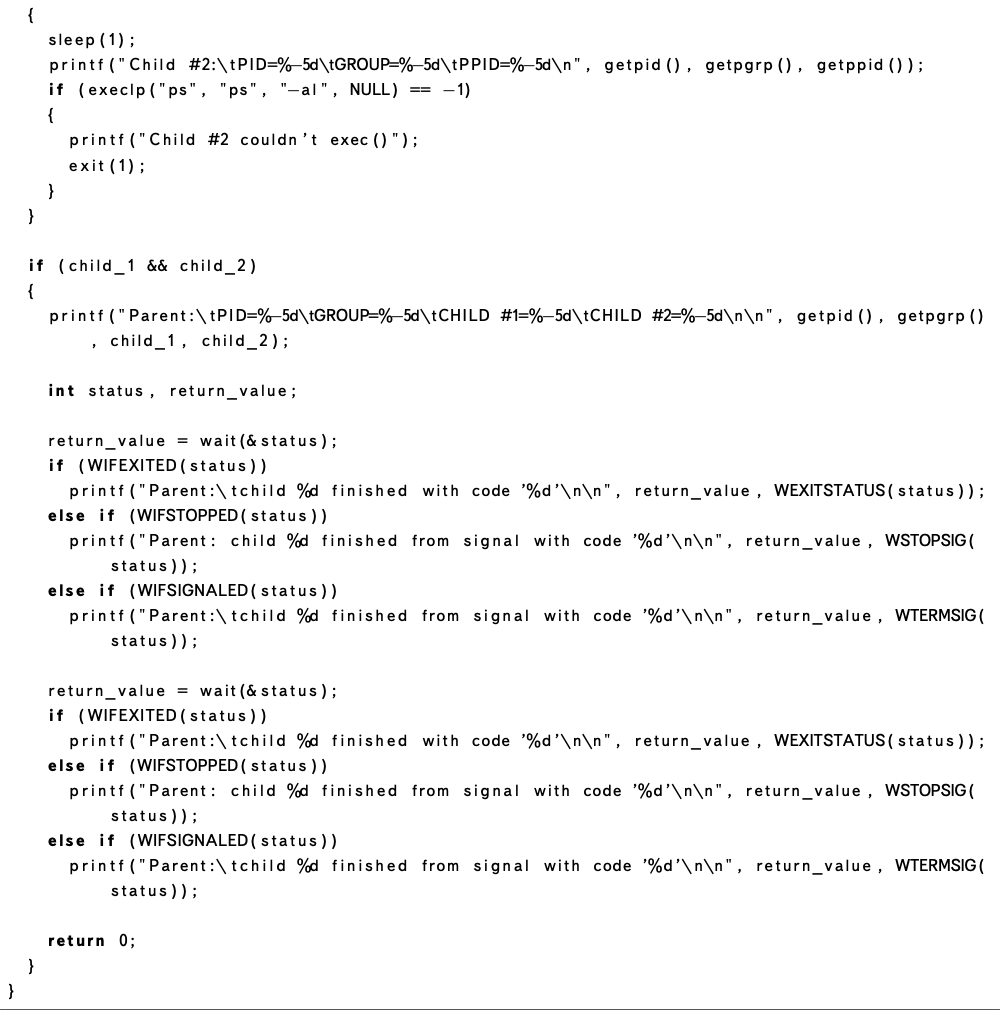
Написать программу, в которой процесс-потомок вызывает систем- ный вызов exec(), а процесс-предок ждет завершения процесса-потомка. Следует создать не менее двух потомков.

В своей программе с помощью системного вызова exec() я запускаю программы ls (ls -a) и ps (ps -al).

Ниже приведён листинг программы для задания №3.

Graphical user interface, text, application, email

Description automatically generated



Ниже приведён пример работы программы для задания №3.

A picture containing text

Description automatically generated

Задание №4

Написать программу, в которой предок и потомок обмениваются со- общением через программный канал.

Ниже приведён листинг программы для задания №4.

A picture containing text

Description automatically generated

Text

Description automatically generated

Ниже приведён пример работы программы для задания №4.

Text

Description automatically generated

Задание №5

В программу с программным каналом включить собственный обра- ботчик сигнала. Использовать сигнал для изменения хода выполнения программы.

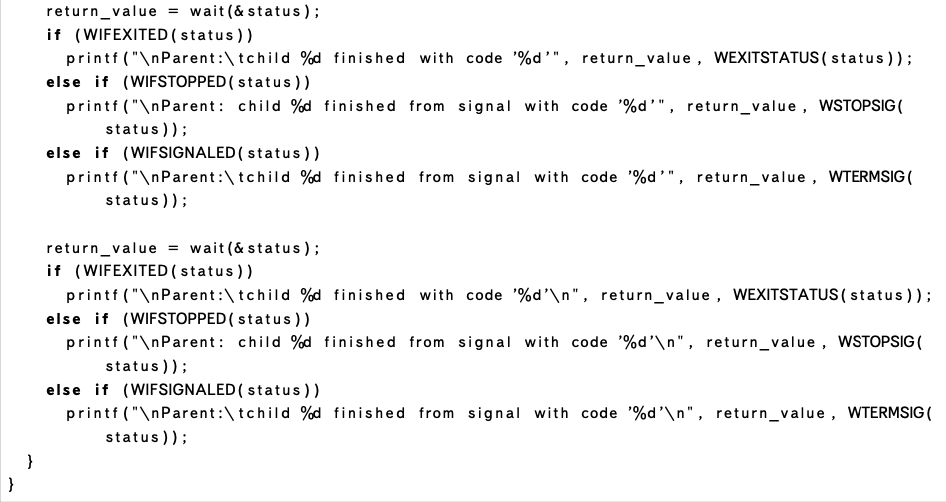
В своей программе я написал обработчики сигналов SIGINT (сиг- нал прерывания (Ctrl-C) с терминала) и SIGALRM (сигнал истечения времени, заданного alarm()).

Ниже приведён листинг программы для задания №5.

Graphical user interface, text

Description automatically generated

Text, application

Description automatically generated



Ниже приведён пример работы программы для задания №5, сигнал прерывания с терминала.

Text

Description automatically generated

Ниже приведён пример работы программы для задания №5, сигнал истечения времени, заданного alarm().

Text

Description automatically generated

Заключение

Проделав лабораторную работу, я познакомился с такими системны- ми вызовами, как fork(), wait(), exec(), pipe() и signal(); изучил как писать программы, используя эти системные вызовы, а также познако- мился с POSIX сигналами.