

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра электронных вычислительных машин

Лабораторная работа № 2  
По дисциплине ОСиСП

«Изучение системных вызовов `fork()`, `execve()`, `getpid()`, `getppid()`, `getenv()`»

Выполнил:  
Студент группы 150501  
Барило К.С.

Проверил:  
старший преподаватель  
\_\_\_\_\_ Поденок Л.П.

Минск 2023

## 1 Постановка задачи

Разработать две программы – parent и child.

Перед запуском программы parent в окружении создается переменная среды CHILD\_PATH с именем каталога, где находится программа child.

Родительский процесс (программа parent) после запуска получает переменные среды, сортирует их в LC\_COLLATE=C и выводит в stdout.

После этого входит в цикл обработки нажатий клавиатуры.

Символ «+», используя fork(2) и execve(2) порождает дочерний процесс и запускает в нем очередной экземпляр программы child. Информацию о каталоге, где размещается child, получает из окружения, используя функцию getenv(). Имя программы (argv[0]) устанавливается как child\_XX, где XX – порядковый номер от 00 до 99. Номер инкрементируется родителем.

Символ «\*» порождает дочерний процесс аналогично предыдущему случаю, однако информацию о расположении программы child получает, сканируя массив параметров среды, переданный в третьем параметре функции main().

Символ «&» порождает дочерний процесс аналогично предыдущему случаю, однако информацию о расположении программы child получает, сканируя массив параметров среды, указанный во внешней переменной extern char \*\*environ, установленной хост-средой при запуске (см. IEEE Std 1003.1-2017).

При запуске дочернего процесса ему передается сокращенное окружение, включающее набор переменных, указанных в файле, который передается родительскому процессу как параметр командной строки.

Минимальный набор переменных должен включать SHELL, HOME, HOSTNAME, LOGNAME, LANG, TERM, USER, LC\_COLLATE, PATH. Дочерний процесс открывает этот файл, считывает имена переменных, получает из окружения их значение и выводит в stdout.

Дочерний процесс (программа child) выводит свое имя, pid, ppid, открывает файл с набором переменных, считывает их имена, получает из окружения, переданного ему при запуске, их значение способом, указанным при обработке нажатий, выводит в stdout и завершается.

Иной символ завершает выполнение родительского процесса.

## 2 Алгоритм

Родительская программа запускается, с помощью итерации по массиву переменных среды env[], переданному через аргумент функции main() сортирует значения переменных среды функцией strcoll(), выводит в stdout.

Далее входит в бесконечный цикл, обрабатывающий вводимые символы в соответствии с заданием. Если введен нужный символ, то вызывается функция, внутри которой с помощью системных вызовов fork() и execle() создается дочерний процесс. В окружение ему передаются переменные, записанные в файле, имя которого передается родителю первым аргументом командной строки.

Дочерний процесс выводит свой pid и ppid, полученное окружение и завершается.

### **3 Тестирование программы**

#### **Ввод:**

```
make
gcc -W -Wall -Wextra -std=c11 parent.c -o parent
gcc -W -Wall -Wextra -std=c11 child.c -o child
./parent ./environ
```

#### **Вывод:**

```
Environment:
CHILD_PATH=./child
COLORTERM=truecolor
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus,guid=2db34ac99dd70f64681e0401641ee3a4
DBUS_STARTER_ADDRESS=unix:path=/run/user/1000/bus,guid=2db34ac99dd70f64681e0401641ee3a4
DBUS_STARTER_BUS_TYPE=session
DESKTOP_SESSION=ubuntu
DISPLAY=:0
GDMSESSION=ubuntu
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GNOME_SETUP_DISPLAY=:1
GNOME_SHELL_SESSION_MODE=ubuntu
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/b855b61f_3ed8_4119_88e0_8e2d46b749a2
GNOME_TERMINAL_SERVICE=:1.100
GTK_MODULES=gail:atk-bridge
HOME=/home/konstantinopol
IM_CONFIG_PHASE=1
LANG=ru_RU.UTF-8
LC_ADDRESS=be_BY.UTF-8
LC_IDENTIFICATION=be_BY.UTF-8
LC_MEASUREMENT=be_BY.UTF-8
LC_MONETARY=be_BY.UTF-8
LC_NAME=be_BY.UTF-8
LC_NUMERIC=be_BY.UTF-8
LC_PAPER=be_BY.UTF-8
LC_TELEPHONE=be_BY.UTF-8
LC_TIME=be_BY.UTF-8
LESSCLOSE=/usr/bin/lesspipe %s %s
LESSOPEN=| /usr/bin/lesspipe %s
LOGNAME=konstantinopol
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc
```

al/games:/snap/bin:/snap/bin  
PWD=/home/konstantinopol/Рабочий стол/UNIXProg/Лабы/lr2  
QT\_ACCESSIBILITY=1  
QT\_IM\_MODULE=ibus  
SESSION\_MANAGER=local/konstantinopol-TUF-Gaming-FX505DT-  
TUF505DT:@/tmp/.ICE-unix/1717,unix/konstantinopol-TUF-Gaming-FX505DT-  
TUF505DT:/tmp/.ICE-unix/1717  
SHELL=/bin/bash  
SHLVL=1  
SSH\_AGENT\_LAUNCHER=gnome-keyring  
SSH\_AUTH\_SOCK=/run/user/1000/keyring/ssh  
SYSTEMD\_EXEC\_PID=1717  
TERM=xterm-256color  
USER=konstantinopol  
USERNAME=konstantinopol  
VTE\_VERSION=6800  
WAYLAND\_DISPLAY=wayland-0  
XAUTHORITY=/run/user/1000/.mutter-Xwaylandauth.H2JB21  
XDG\_CONFIG\_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg  
XDG\_CURRENT\_DESKTOP=ubuntu:GNOME  
XDG\_DATA\_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/var/lib/snapd/d  
esktop  
XDG\_MENU\_PREFIX=gnome-  
XDG\_RUNTIME\_DIR=/run/user/1000  
XDG\_SESSION\_CLASS=user  
XDG\_SESSION\_DESKTOP=ubuntu  
XDG\_SESSION\_TYPE=wayland  
XMODIFIERS=@im=ibus

Choose one of: '\*', '+', '&' (default - exit from program))

&

Start process & ...

Process CHILD\_0 start

pid = 5137

ppid = 5136

File path = './environ'

SHELL=/bin/bash

HOME=/home/konstantinopol

LOGNAME=konstantinopol

LANG=ru\_RU.UTF-8

TERM=xterm-256color

USER=konstantinopol

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/loc  
al/games:/snap/bin:/snap/bin

Костянчик, [25.03.2023 9:56]

Child process finished with code 0

Choose one of: '\*', '+', '&' (default - exit from program))

h

#### **4 Вывод**

В ходе выполнения лабораторной работы были изучены основы работы с переменными среды и освоены системные вызовы `fork()` и `execve()`.