

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин

Лабораторная работа №2
по курсу
«Операционные системы и системное программирование»
на тему
«Понятие процессов»

Выполнил:

студент группы 350501

Проверил:

Русак Г.Д.
старший преподаватель каф. ЭВМ
Поденок Л.П.

МИНСК 2025

1 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1.1 Условие лабораторной работы

Разработать две программы – `parent` (родительский процесс) и `child` (дочерний процесс).

Родительский процесс, запуская дочерний, создает для него сокращенную среду (окружение). Для этого пользователем создается файл `env`, содержащий небольшой набор имен переменных окружения, передаваемых при вызове `execve()`.

Минимальный набор переменных в файле `env` должен включать `SHELL`, `HOME`, `HOSTNAME`, `LOGNAME`, `LANG`, `TERM`, `USER`, `LC_COLLATE`, `PATH`.

Перед запуском программы `parent` в ее окружении пользователем создается переменная `CHILD_PATH` с именем каталога, где находится программа `child`.

Родительский процесс (программа `parent`) после запуска получает переменные своего окружения и их значения, установленные оболочкой, сортирует в `LC_COLLATE=C` и выводит в `stdout`. Читает файл `env` и формирует среду для дочернего процесса в том виде, в котором она указывается в системном вызове `execve()`, используя значения для переменных из собственной среды. После этого входит в цикл обработки нажатий клавиатуры.

1.1.1 Символ «+»

Родительский процесс, используя `fork(2)` и `execve(2)` порождает дочерний процесс и запускает в нем очередной экземпляр программы `child`. Информацию о каталоге, где размещается `child`, `parent` получает из своего окружения, используя функцию `getenv()`.

Имя программы `child` (`argv[0]`) устанавливается как `child_XX`, где `XX` – порядковый номер от 00 до 99 (номер инкрементируется родителем). Дочерний процесс выводит свое имя, `pid` и `ppid` в `stdout`. Вторым параметром программы `child` является путь к файлу `env`, который читается дочерним процессом для получения ему переданных значений параметров среды. Дочерний процесс открывает этот файл, считывает имена переменных, получает из окружения их значение, используя `getenv()`, и выводит в `stdout`.

1.1.2 Символ «*»

Дочерний процесс порождается аналогично предыдущему случаю, однако информацию о своем окружении программа `child` получает, сканируя массив параметров среды, переданный в третьем параметре функции `main()` и выводит в `stdout`. Путь к файлу `env` передавать в параметрах не требуется.

1.1.3 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

1.1.4 Символ «q»

Завершает выполнение родительского процесса после завершения дочернего.

1.2 Требования к сборке

Программы компилируются с ключами

`-W -Wall -Wextra -std=c11 -pedantic`

Допускается использование ключей

`-Wno-unused-parameter -Wno-unused-variable`.

Для компиляции, сборки и очистки используется `make`.

2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 Описание алгоритма выполнения работы

Программа предназначена для создания и управления дочерними процессами с использованием системных вызовов `fork()` и `execve()`. Родительский процесс формирует сокращенное окружение на основе файла `env`, сортирует переменные окружения. В зависимости от ввода пользователя (+, *, q), родительский процесс создает дочерние процессы, передавая им окружение либо через аргументы командной строки, либо через массив `envp`, либо через массив параметров внешней среды `environ`, и завершает работу при вводе символа q.

Дочерний процесс выводит информацию о себе (имя, `pid`, `ppid`) и значения переменных окружения, полученных из файла `env` или переданного массива `envp`. Программа использует переменную `CHILD_PATH` для определения пути к дочернему процессу, обеспечивая корректное взаимодействие между процессами.

2.2 Функциональная структура проекта

Файл `env` — набор переменных окружения, которые будут переданы дочернему процессу при его запуске. Данный набор включает следующее:

- 1) `SHELL` — путь к текущей оболочке;
- 2) `HOME` — домашний каталог пользователя;
- 3) `HOSTNAME` — имя хоста системы ;
- 4) `LOGNAME` — имя пользователя для входа в систему;
- 5) `LANG` — текущая языковая локаль;
- 6) `TERM` — тип терминала;
- 7) `USER` — имя текущего пользователя ;
- 8) `LC_COLLATE` — настройки сортировки символов;
- 9) `PATH` — пути для поиска исполняемых файлов.

Файл `child.c` содержит программу `child`, которая представляет собой дочерний процесс, который выводит информацию о себе (имя, `PID`, `PPID`) и обрабатывает окружение в зависимости от переданных аргументов.

Основные функции в `child.c`:

`print_env_from_file()` — выводит переменные окружения, указанные в файле. Принимаемые параметры:

1) `const char *filename` — имя файла, содержащего список переменных окружения.

`print_all_env()` — выводит все переменные окружения, переданные через `envp`. Принимаемые параметры:

1) `char *envp[]` — массив строк, содержащий переменные окружения.

`main()` — основная функция программы, которая выводит информацию о процессе. Принимаемые параметры:

- 1) `int argc` — количество аргументов командной строки;
- 2) `char* argv[]` — массив аргументов командной строки;
- 3) `char* envp[]` — массив переменных окружения.

Файл `parent.c` содержит программу `parent`, которая предназначена для работы с переменными окружения и создания дочерних процессов.

Основные функции в `parent.c`:

`compare_strings()` — сравнивает передаваемые строки для сортировки.

Функция возвращает число, которое является результатом сравнения двух строк.

Принимаемые параметры:

- 1) `const void *a` — указатель на первый элемент для сравнения;
- 2) `const void* b` — указатель на второй элемент для сравнения.

`create_child_env()` — создает окружение для дочернего процесса на основе файла. Функция возвращает массив строк, представляющих окружение.

`main()` — основная функция родительского процесса. Функция в зависимости от ввода пользователя запускает дочерние процессы либо с специально созданным окружением, либо окружение родительского процесса. Принимаемые параметры аналогичны функции `main` для дочернего процесса.

2.3 Порядок сборки и использования

Порядок сборки и запуска состоит в следующем:

- 1) Клонировать репозиторий, используя команду
`git clone https://github.com/Everolfe/parent-child_OSASP`,
или разархивировать каталог с проектом;
- 2) Перейти в каталог с проектом
`cd parent-child_OSASP`,
или
`cd "Имя разархивированного каталога"`;
- 3) Собрать проект используя `make`;
- 4) Установить переменную окружения `CHILD_PATH`
`export CHILD_PATH=$(pwd)`
- 5) После сборки проекта можно использовать, прописав
`./parent`.

3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

Запуск программы:

```
grusak@fedora:~/tar_working_dir/Пысак Г.Д./lab02$ ./parent
CHILD_PATH=/home/grusak/tar_working_dir/Пысак Г.Д./lab02
COLORTERM=truecolor
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
DEBUGINFOD_IMA_CERT_PATH=/etc/keys/ima:
DEBUGINFOD_URLS=https://debuginfod.fedoraproject.org/
DESKTOP_SESSION=xfce
DISPLAY=:0.0
EDITOR=/usr/bin/nano
GDMSESSION=xfce
GDM_LANG=ru_RU.utf8
GPG_TTY=/dev/pts/2
GTK_IM_MODULE=xim
HISTCONTROL=ignoredups
HISTSIZE=1000
HOME=/home/grusak
HOSTNAME=fedora
IMSETTINGS_INTEGRATE_DESKTOP=yes
IMSETTINGS_MODULE=X compose table
LANG=ru_RU.utf8
LC_COLLATE=C
LESSOPEN=||/usr/bin/lesspipe.sh %s
LOGNAME=grusak
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=
01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;37;41:su=37
;41:sg=30;43:ca=00:tw=30;42:ow=34;42:st=37;44:ex=01;32:*
.7z=01;31:*
.ace=01;31:*
.alz=01;31:*
.apk=01;31:*
.arc=01;31:*
.ar
j=01;31:*
.bz=01;31:*
.bz2=01;31:*
.cab=01;31:*
.cpio=01;31:*
.c
rate=01;31:*
.deb=01;31:*
.drpm=01;31:*
.dwm=01;31:*
.dz=01;31:
*
.ear=01;31:*
.egg=01;31:*
.esd=01;31:*
.gz=01;31:*
.jar=01;31:
*
.lha=01;31:*
.lrz=01;31:*
.lz=01;31:*
.lz4=01;31:*
.lzh=01;31:
*
.lzma=01;31:*
.lzo=01;31:*
.pyz=01;31:*
.rar=01;31:*
.rpm=01;3
1:*
.rz=01;31:*
.sar=01;31:*
.swm=01;31:*
.t7z=01;31:*
.tar=01;3
1:*
.taz=01;31:*
.tbz=01;31:*
.tbz2=01;31:*
.tgz=01;31:*
.tlz=01
;31:*
.txz=01;31:*
.tz=01;31:*
.tzo=01;31:*
.tzst=01;31:*
.udeb=
01;31:*
.war=01;31:*
.whl=01;31:*
.wim=01;31:*
.xz=01;31:*
.z=01
;31:*
.zip=01;31:*
.zoo=01;31:*
.zst=01;31:*
.avif=01;35:*
.jpg=
01;35:*
.jpeg=01;35:*
.mjpg=01;35:*
.mjpeg=01;35:*
.gif=01;35:*
.
.bmp=01;35:*
.pbm=01;35:*
.pgm=01;35:*
.ppm=01;35:*
.tga=01;35:
*
.xbm=01;35:*
.xpm=01;35:*
.tif=01;35:*
.tiff=01;35:*
.png=01;3
5:*
.svg=01;35:*
.svgz=01;35:*
.mng=01;35:*
.pcx=01;35:*
.mov=01
;35:*
.mpg=01;35:*
.mpeg=01;35:*
.m2v=01;35:*
.mkv=01;35:*
.webm
=01;35:*
.webp=01;35:*
.ogm=01;35:*
.mp4=01;35:*
.m4v=01;35:*
.m
p4v=01;35:*
.vob=01;35:*
.qt=01;35:*
.nuv=01;35:*
.wmv=01;35:*
.
.asf=01;35:*
.rm=01;35:*
.rmvb=01;35:*
.flc=01;35:*
.avi=01;35:*
.
.fli=01;35:*
.flv=01;35:*
.gl=01;35:*
.dl=01;35:*
.xcf=01;35:*
.
```

xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.m4a=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.oga=01;36:*.opus=01;36:*.spx=01;36:*.xspf=01;36:*~=00;90:*#=00;90:*.bak=00;90:*.crdownload=00;90:*.dpgk-dist=00;90:*.dpgk-new=00;90:*.dpgk-old=00;90:*.dpgk-tmp=00;90:*.old=00;90:*.orig=00;90:*.part=00;90:*.rej=00;90:*.rpmnew=00;90:*.rpmorig=00;90:*.rpmsave=00;90:*.swp=00;90:*.tmp=00;90:*.ucf-dist=00;90:*.ucf-new=00;90:*.ucf-old=00;90:
MAIL=/var/spool/mail/grusak
MOZ_GMP_PATH=/usr/lib64/mozilla/plugins/gmp-gmpopenh264/system-installed
OLDPWD=/home/grusak
PATH=/home/grusak/.local/bin:/home/grusak/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
PWD=/home/grusak/tar_working_dir/Рысак Р.Д./lab02
QT_IM_MODULE=xim
SESSION_MANAGER=local/unix:@/tmp/.ICE-unix/1455,unix/unix:/tmp/.ICE-unix/1455
SHELL=/bin/bash
SHLVL=1
SSH_AGENT_PID=1630
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
SSH_AUTH_SOCK=/tmp/ssh-XXXXXXj1kymQ/agent.1455
TERM=xterm-256color
USER=grusak
VTE_VERSION=7802
WINDOWID=31457333
XAUTHORITY=/run/lightdm/grusak/xauthority
XDG_CACHE_HOME=/home/grusak/.cache
XDG_CONFIG_DIRS=/etc/xdg
XDG_CONFIG_HOME=/home/grusak/.config
XDG_CURRENT_DESKTOP=XFCE
XDG_DATA_DIRS=/usr/local/share:/usr/share
XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/grusak
XDG_MENU_PREFIX=xfce-
XDG_RUNTIME_DIR=/run/user/1000
XDG_SEAT=seat0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
XDG_SESSION_CLASS=user
XDG_SESSION_DESKTOP=xfce
XDG_SESSION_ID=2
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0
XDG_SESSION_TYPE=x11
XDG_VTNR=1
XMODIFIERS=@im=none
_=./parent

Ввод символа «+»:

Menu:

- + - Launch child with getenv() path
- * - Launch child with envp path
- & - Launch child with environ path
- q - Quit

Enter command: +

Имя процесса: child_00

PID: 3540

PPID: 3491

SHELL=/bin/bash

HOME=/home/grusak

HOSTNAME=fedora

LOGNAME=grusak

LANG=ru_RU.utf8

TERM=xterm-256color

USER=grusak

LC_COLLATE=C

PATH=/home/grusak/.local/bin:/home/grusak/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

Ввод символа «*»:

Menu:

- + - Launch child with getenv() path
- * - Launch child with envp path
- & - Launch child with environ path
- q - Quit

Enter command: *

Имя процесса: child_01

PID: 3600

PPID: 3491

SHELL=/bin/bash

HOME=/home/grusak

HOSTNAME=fedora

LOGNAME=grusak

LANG=ru_RU.utf8

TERM=xterm-256color

USER=grusak

LC_COLLATE=C

PATH=/home/grusak/.local/bin:/home/grusak/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

Ввод символа «&»:

Menu:

- + - Launch child with getenv() path
- * - Launch child with envp path
- & - Launch child with environ path
- q - Quit


```
Enter command: &  
Имя процесса: child_02  
PID: 3614  
PPID: 3491  
SHELL=/bin/bash  
HOME=/home/grusak  
HOSTNAME=fedora  
LOGNAME=grusak  
LANG=ru_RU.utf8  
TERM=xterm-256color  
USER=grusak  
LC_COLLATE=C  
PATH=/home/grusak/.local/bin:/home/grusak/bin:/usr/local/  
bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

Ввод символа «q»:

Menu:

```
+ - Launch child with getenv() path  
* - Launch child with envp path  
& - Launch child with environ path  
q - Quit
```

Enter command: q

Ввод некорректного символа:

Menu:

```
+ - Launch child with getenv() path  
* - Launch child with envp path  
& - Launch child with environ path  
q - Quit
```

Enter command: a

Invalid command