

Министерство образования Республики Беларусь  
Учреждение Образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных машин

Лабораторная работа № 2  
«Понятие процессов»

Выполнил:

студент группы 350501

Слепица О.Н.

Проверил:

старший преподаватель каф. ЭВМ

Поденок Л.П.

Минск 2025

## 1 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

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.

1) 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.

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

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.

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.

Символ «q» завершает выполнение родительского процесса.

Требования к сборке:

- 1) Программы компилируются с ключами  
`-W -Wall -Wextra -std=c11 -pedantic`  
`-Wno-unused-parameter -Wno-unused-variable;`
- 2) Для компиляции, сборки и очистки используется `make`.

## 2 ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ

### 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 `envr`, 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 `envr`. 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.

Рассмотрим основные функции в `child.c`.

Функция `display_env_from_config()` выводит значения переменных окружения, перечисленных в указанном файле. Принимаемые параметры:

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

Функция `display_environment_vars()` выводит все переменные окружения, переданные процессу.

Принимаемые параметры:

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

Функция `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.

Рассмотрим основные функции в `parent.c`.

Функция `string_comparator()` сравнивает передаваемые строки для сортировки. Функция возвращает число, которое является результатом сравнения двух строк. Принимаемые параметры:

1) `const void *first` — указатель на первую строку;

2) `const void* second` — указатель на вторую строку.

Функция `generate_process_environment()` создает окружение для дочернего процесса на основе файла. Функция возвращает массив строк, представляющих окружение. Принимаемые параметры:

1) `env_config_file (const char*)` — путь к файлу.

Функция `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:

1) `int argc` — количество аргументов командной строки;

2) `char* argv[]` — массив аргументов командной строки;

3) `char* envp[]` — массив переменных окружения.

## 2.3 Порядок сборки и использования

Порядок сборки и запуска состоит в следующем:

1) Разархивировать каталог с проектом;

2) Перейти в каталог с проектом `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 ТЕСТИРОВАНИЯ

```
helga@fedora:~/tar_working_dir/Слепица 0.Н./lab02$ ./parent
child_path
CHILD_PATH=/home/helga/tar_working_dir/Слепица 0.Н./lab02
COLORFGBG=15;0
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=plasma
DISPLAY=:0
EDITOR=/usr/bin/nano
GDK_CORE_DEVICE_EVENTS=1
GPG_TTY=/dev/pts/1
GTK2_RC_FILES=/home/helga/.gtkrc-2.0-kde4
GTK_RC_FILES=/etc/gtk/gtkrc:/home/helga/.gtkrc:/home/
helga/.config/gtkrc
HISTCONTROL=ignoredups
HISTSIZE=1000
HOME=/home/helga
HOSTNAME=fedora
ICEAUTHORITY=/run/user/1000/iceauth_OIJqti
IMSETTINGS_INTEGRATE_DESKTOP=yes
IMSETTINGS_MODULE=X compose table
INVOCATION_ID=3d93d44dc40e444d95e16bc6ebdb3b33
JOURNAL_STREAM=9:21544
KDEDIRS=/usr
KDE_APPLICATIONS_AS_SCOPE=1
KDE_FULL_SESSION=true
KDE_SESSION_UID=1000
KDE_SESSION_VERSION=6
KONSOLE_DBUS_SERVICE=:1.151
KONSOLE_DBUS_SESSION=/Sessions/1
KONSOLE_DBUS_WINDOW=/Windows/1
KONSOLE_VERSION=241202
LANG=ru_RU.UTF-8
LANGUAGE=
LC_COLLATE=C
LESSOPEN=||/usr/bin/lesspipe.sh %s
LOGNAME=helga
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:*.arj=01;31:*.bz=01;31:*.bz2=0
1;31:*.cab=01;31:*.cpio=01;31:*.crate=01;31:*.deb=01;31:*.drpm=01;3
1:*.dwm=01;31:*.dz=01;31:*.ear=01;31:*.egg=01;31:*.esd=01;31:*.gz=0
1;31:*.jar=01;31:*.lha=01;31:*.lrz=01;31:*.lz=01;31:*.lz4=01;31:*.l
zh=01;31:*.lzma=01;31:*.lzo=01;31:*.pyz=01;31:*.rar=01;31:*.rpm=01;
31:*.rz=01;31:*.sar=01;31:*.swm=01;31:*.t7z=01;31:*.tar=01;31:*.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:*.z
st=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;35:*.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:*.web
p=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;3
5:*.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;3
5:*.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
:*.dpkg-dist=00;90:*.dpkg-new=00;90:*.dpkg-old=00;90:*.dpgk-
tmp=00;90:*.old=00;90:*.orig=00;90:*.part=00;90:*.rej=00;90:*.rpmne
w=00;90:*.rpmorig=00;90:*.rpmsave=00;90:*.swp=00;90:*.tmp=00;90:*.u
cf-dist=00;90:*.ucf-new=00;90:*.ucf-old=00;90:
MAIL=/var/spool/mail/helga
MANAGERPID=1374
MEMORY_PRESSURE_WATCH=/sys/fs/cgroup/user.slice/
user-1000.slice/user@1000.service/session.slice/plasma-
plasmashell.service/memory.pressure
MEMORY_PRESSURE_WRITE=c29tZSAyMDAwMDAgMjAwMDAwMAA=
MOZ_GMP_PATH=/usr/lib64/mozilla/plugins/gmp-gmpopenh264/
system-installed
PAM_KWALLET5_LOGIN=/run/user/1000/kwallet5.socket
PATH=/home/helga/.local/bin:/home/helga/bin:/usr/local/bin:/
usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
PROFILEHOME=
PWD=/home/helga/tar_working_dir/Слепица О.Н./lab02
QT_AUTO_SCREEN_SCALE_FACTOR=0
QT_IM_MODULE=xim
QT_WAYLAND_RECONNECT=1
SESSION_MANAGER=local/unix:~/tmp/.ICE-unix/1780,unix/unix:/
tmp/.ICE-unix/1780
SHELL=/bin/bash
SHELL_SESSION_ID=f6c415db6e024419b146904a08094750
SHLVL=1
SSH_ASKPASS=/usr/bin/ksshaskpass
SSH_AUTH_SOCK=/run/user/1000/ssh-agent.socket
SYSTEMD_EXEC_PID=1836
SYSTEMD_SLEEP_FREEZE_USER_SESSIONS=0
TERM=xterm-256color
USER=helga
WAYLAND_DISPLAY=wayland-0
WINDOWID=3
XAUTHORITY=/run/user/1000/xauth_THftJN

```

```
XDG_CONFIG_DIRS=/home/helga/.config/kdedefaults:/etc/xdg:/usr/share/kde-settings/kde-profile/default/xdg
XDG_CURRENT_DESKTOP=KDE
XDG_DATA_DIRS=/home/helga/.local/share/flatpak/exports/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/share
XDG_MENU_PREFIX=plasma-
XDG_RUNTIME_DIR=/run/user/1000
XDG_SEAT=seat0
XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0
XDG_SESSION_CLASS=user
XDG_SESSION_DESKTOP=KDE
XDG_SESSION_ID=2
XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session1
XDG_SESSION_TYPE=wayland
XDG_VTNR=2
XKB_DEFAULT_LAYOUT=us,ru
XKB_DEFAULT_MODEL=pc105
XKB_DEFAULT_OPTIONS=grp:alt_shift_toggle
XKB_DEFAULT_VARIANT=,
XMODIFIERS=@im=none
_=./parent
```

Options Menu:

```
+ - Start process using getenv() path
* - Start process using envp path
& - Start process using environ path
q - Quit program
```

Your selection: +

Process Name: child\_00

Process ID: 8536

Parent Process ID: 8535

SHELL=/bin/bash

HOME=/home/helga

HOSTNAME=fedora

LOGNAME=helga

LANG=ru\_RU.UTF-8

TERM=xterm-256color

USER=helga

LC\_COLLATE=C

```
PATH=/home/helga/.local/bin:/home/helga/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
```

Options Menu:

```
+ - Start process using getenv() path
* - Start process using envp path
& - Start process using environ path
q - Quit program
```

Your selection: \*

Process Name: child\_01

Process ID: 8538

Parent Process ID: 8535

```
SHELL=/bin/bash
HOME=/home/helga
HOSTNAME=fedora
LOGNAME=helga
LANG=ru_RU.UTF-8
TERM=xterm-256color
USER=helga
LC_COLLATE=C
PATH=/home/helga/.local/bin:/home/helga/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
```

Options Menu:

```
+ - Start process using getenv() path
* - Start process using envp path
& - Start process using environ path
q - Quit program
```

Your selection: &

Process Name: child\_02

Process ID: 8543

Parent Process ID: 8535

SHELL=/bin/bash

HOME=/home/helga

HOSTNAME=fedora

LOGNAME=helga

LANG=ru\_RU.UTF-8

TERM=xterm-256color

USER=helga

LC\_COLLATE=C

```
PATH=/home/helga/.local/bin:/home/helga/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
```

Options Menu:

```
+ - Start process using getenv() path
* - Start process using envp path
& - Start process using environ path
q - Quit program
```

Your selection: K

Invalid option selected

Options Menu:

```
+ - Start process using getenv() path
* - Start process using envp path
& - Start process using environ path
q - Quit program
```

Your selection: q