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

Кафедра электронных вычислительных машин

Лабораторная работа № 2
«Понятие процессов.»

Проверил: Поденок Л.П.
Выполнил: ст. гр. 350501
Маслаков Н.А.

Минск 2025

1. УСЛОВИЕ ЛАБАРАТОРНОЙ 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.

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.

Символ «+»

Родительский процесс, используя `fork()` и `execve()` 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`.

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.

Символ «*»

Дочерний процесс порождается аналогично предыдущему случаю, однако информацию о своем окружении программа `child` получает, сканируя глобальный массив `environ`.

Символ «q»

Завершает выполнение родительского процесса после завершения дочернего.

2. ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ

В программе родительский процесс, создавая сокращенную среду, порождает дочерний. В начале работы она инициализирует локаль для корректной сортировки. Затем выводит все переменные окружения из массива `environ`. После этого, в зависимости от выбранной команды запуска, порождается дочерний процесс с сокращенной средой.

3. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА

Родительский процесс.

`int EnvCmp(const void* a, const void* b).`

Функция `EnvCmp` является компоратором, используемым для функции `qsort` для сортировки переменных окружения.

Принимаемые параметры:

1) `const void* a`. Первая строка.

2) `const void* b`. Вторая строка

`void PrintEnvSorted().`

Функция `EnvCmp` используется для вывода отсортированного с помощью `qsort` окружения родительского процесса.

`char** CreateChildEnv().`

Функция `CreateChildEnv` используется для создания сокращённого окружения для дочернего процесса. Переменные сокращенного окружения считываются из файла `env.txt`.

`char FindChildPath(const char mode, char** child_env).`

Функция `FindChildPath` используется для поиска пути к программе `child` в зависимости от режима(+, *, &).

Принимаемые параметры:

1) `const char mode`. Режим получения переменных окружения.

2) `char** child_env`. Массив с переменными сокращенного окружения.

`void StartChild(const char mode, char** child_env).`

Функция `FindChildPath` используется для создания дочернего процесса с сокращенной средой.

Принимаемые параметры:

1) `const char mode`. Режим получения переменных окружения.

2) `char** child_env`. Массив с переменными сокращенного окружения.

Функция `main` является основной точкой входа в программу, которая реализует функциональность вывода всех переменных окружения, создания массива сокращенной среды, а также запуска дочернего процесса.

Дочерний процесс.

`int main(int argc, char* argv[], char* envp).`

Функция `main` является основной точкой входа в программу, которая реализует функциональность вывода базовой информации о процессе и выбора, откуда брать переменные окружения.

4. ПОРЯДОК СБОРКИ И ЗАПУСКА

1) Перейти в каталог проекта.

```
$ cd 'Маслаков Н.А./lab02'
```

2) Собрать проект с помощью `make`. по умолчанию сборка происходит в режиме отладки.

```
$ make
```

3) Установить переменную окружения `CHILD_PATH`

```
$ export CHILD_PATH=./build/debug/child
```

4) Запустить программу.

```
$ ./build/debug/parent
```

5. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

```
~/lab02$ ./build/debug/parent
Родительское окружение (отсортированное):
CHILD_PATH=./build/debug/child
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
GDMSESSION=plasma
GDM_LANG=ru_RU.UTF-8
GPG_TTY=/dev/pts/2
GTK2_RC_FILES=/home/hechert/.gtkrc-2.0-kde4
GTK_RC_FILES=/etc/gtk/gtkrc:/home/hechert/.gtkrc:/home/
hechert/.config/gtkrc
HISTCONTROL=ignoreboth
HOME=/home/hechert
HOSTNAME=fedora
ICEAUTHORITY=/run/user/1000/iceauth_HLRfBt
INVOCATION_ID=59ae21ba38da42ae994cf4385cbce25f
JOURNAL_STREAM=9:29952
KDEDIRS=/usr
KDE_APPLICATIONS_AS_SCOPE=1
KDE_FULL_SESSION=true
KDE_SESSION_UID=1000
KDE_SESSION_VERSION=6
KGLOBALACCELD_PLATFORM=org.kde.kwin
KONSOLE_DBUS_SERVICE=:1.84
KONSOLE_DBUS_SESSION=/Sessions/1
KONSOLE_DBUS_WINDOW=/Windows/1
KONSOLE_VERSION=241203
LANG=ru_RU.UTF-8
LANGUAGE=
LC_COLLATE=C
LESSOPEN=||/usr/bin/lesspipe.sh %s
LOGNAME=hechert
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=01;31:*.cab=0
1;31:*.cpio=01;31:*.crate=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;31:*.rz=01;31:*.sar=01;3
1:*.swm=01;31:*.t7z=01;31:*.tar=01;31:*.taz=01;31:*.tbz=01;31:*.tbz2=0
1;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;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:*.webp=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=
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:*.g
l=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.e
mf=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;3
6:*.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:*.dp
kg-dist=00;90:*.dpkg-new=00;90:*.dpkg-old=00;90:*.dpkg-
tmp=00;90:*.old=00;90:*.orig=00;90:*.part=00;90:*.rej=00;90:*.rpmnew=0
0;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:
     MANAGERPID=2179
     MC_SID=3848
     MC_TMPDIR=/var/tmp/mc-CYQ432
     MEMORY_PRESSURE_WATCH=/sys/fs/cgroup/user.slice/user-1000.slice/
user@1000.service/session.slice/plasma-kwin_wayland.service/
memory.pressure
     MEMORY_PRESSURE_WRITE=c29tZSAyMDAwMDAgMjAwMDAwMAA=
     MOZ_GMP_PATH=/usr/lib64/mozilla/plugins/gmp-gmpopenh264/system-
installed
     OLDPWD=/home/hechert/tar_working_dir/Маслаков Н.А.
     PATH=/home/hechert/.local/bin:/home/hechert/bin:/usr/local/bin:/
usr/local/sbin:/usr/bin:/usr/sbin
     PROFILEHOME=
     PWD=/home/hechert/tar_working_dir/Маслаков Н.А./lab02
     QT_AUTO_SCREEN_SCALE_FACTOR=0
     QT_WAYLAND_RECONNECT=1
     SESSION_MANAGER=local/unix:@/tmp/.ICE-unix/2431,unix/unix:/
tmp/.ICE-unix/2431
     SHELL=/bin/bash
     SHELL_SESSION_ID=9bd4e109a883416a99f273e2bea65bde
     SHLVL=2
     SSH_ASKPASS=/usr/bin/ksshaskpass
     SSH_AUTH_SOCK=/run/user/1000/ssh-agent.socket
     SYSTEMD_EXEC_PID=2255
     SYSTEMD_SLEEP_FREEZE_USER_SESSIONS=0
     TERM=xterm-256color
     USER=hechert
     USERNAME=hechert
     WAYLAND_DISPLAY=wayland-0
     WINDOWID=1
     XAUTHORITY=/run/user/1000/xauth_NXsFez
     XDG_CONFIG_DIRS=/home/hechert/.config/kdedefaults:/etc/xdg:/usr/
share/kde-settings/kde-profile/default/xdg
     XDG_CURRENT_DESKTOP=KDE
     XDG_DATA_DIRS=/home/hechert/.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_SESSION_CLASS=user
      XDG_SESSION_DESKTOP=plasma
      XDG_SESSION_ID=2
      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=,
      _=./build/debug/parent

      Введите команду:
        '+' : запустить child с чтением переменных из файла
        '*' : запустить child с выводом окружения, переданного через
envp
        '&' : запустить child с поиском CHILD_PATH в глобальном
окружении
        'q' : завершить работу программы
      +
      Режим '+': Получаем CHILD_PATH через getenv() ->
./build/debug/child
      Имя процесса: child_00
      PID: 3884
      PPID: 3883

      Режим: чтение переменных из файла env.txt через getenv()
      SHELL=/bin/bash
      HOME=/home/hechert
      HOSTNAME=fedora
      LOGNAME=hechert
      LANG=ru_RU.UTF-8
      TERM=xterm-256color
      USER=hechert
      LC_COLLATE=C
      PATH=/home/hechert/.local/bin:/home/hechert/bin:/usr/local/bin:/
usr/local/sbin:/usr/bin:/usr/sbin
      Введите команду (+, *, &, q): *
      Режим '*': Найден CHILD_PATH в envp -> ./build/debug/child
      Имя процесса: child_01
      PID: 3885
      PPID: 3883

      Режим: вывод переменных из переданного окружения (envp):
      SHELL=/bin/bash
      HOME=/home/hechert
      HOSTNAME=fedora
      LOGNAME=hechert
      LANG=ru_RU.UTF-8
      TERM=xterm-256color
      USER=hechert
      LC_COLLATE=C
```

```
     PATH=/home/hechert/.local/bin:/home/hechert/bin:/usr/local/bin:/
usr/local/sbin:/usr/bin:/usr/sbin
     CHILD_PATH=./build/debug/child
     Введите команду (+, *, &, q): &
     Режим '&': Найден CHILD_PATH в глобальном окружении ->
./build/debug/child
     Имя процесса: child_02
     PID: 3886
     PPID: 3883

     Режим: вывод переменных из переданного окружения (envp):
     SHELL=/bin/bash
     HOME=/home/hechert
     HOSTNAME=fedora
     LOGNAME=hechert
     LANG=ru_RU.UTF-8
     TERM=xterm-256color
     USER=hechert
     LC_COLLATE=C
     PATH=/home/hechert/.local/bin:/home/hechert/bin:/usr/local/bin:/
usr/local/sbin:/usr/bin:/usr/sbin
     CHILD_PATH=./build/debug/child
     Введите команду (+, *, &, q): q
```