

Programy v prostředí operačního systému

Programovací techniky

doc. Ing. Jiří Rybička, Dr.
ústav informatiky
PEF MENDELU v Brně
rybicka@mendelu.cz

Standardní soubory

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`
- Soubory není třeba deklarovat a nastavovat

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`
- Soubory není třeba deklarovat a nastavovat
- Specifické odchylky chování při vstupu z klávesnice

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`
- Soubory není třeba deklarovat a nastavovat
- Specifické odchylky chování při vstupu z klávesnice
- Speciální textový soubor `stderr`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`
- Soubory není třeba deklarovat a nastavovat
- Specifické odchylky chování při vstupu z klávesnice
- Speciální textový soubor `stderr`
- Z hlediska OS: možné přesměrování vstupu a výstupu, kolona

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`
- Soubory není třeba deklarovat a nastavovat
- Specifické odchylky chování při vstupu z klávesnice
- Speciální textový soubor `stderr`
- Z hlediska OS: možné přesměrování vstupu a výstupu, kolona
- Efektivní ladění s připravenými daty v souboru

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Představován textovými soubory `input` a `output`
- Soubory není třeba deklarovat a nastavovat
- Specifické odchylky chování při vstupu z klávesnice
- Speciální textový soubor `stderr`
- Z hlediska OS: možné přesměrování vstupu a výstupu, kolona
- Efektivní ladění s připravenými daty v souboru
- Možnost dalšího filtrování nebo zpracování výstupů

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry
- Parametry jsou vhodným doplňkovým vstupem

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry
- Parametry jsou vhodným doplňkovým vstupem
- Obvykle používány na konfiguraci nebo úpravu činnosti programu

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry
- Parametry jsou vhodným doplňkovým vstupem
- Obvykle používány na konfiguraci nebo úpravu činnosti programu
- Funkce pro zjištění počtu parametrů: `ParamCount`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry
- Parametry jsou vhodným doplňkovým vstupem
- Obvykle používány na konfiguraci nebo úpravu činnosti programu
- Funkce pro zjištění počtu parametrů: `ParamCount`
- Funkce pro zjištění hodnoty n -tého parametru:
`ParamStr(n)`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry
- Parametry jsou vhodným doplňkovým vstupem
- Obvykle používány na konfiguraci nebo úpravu činnosti programu
- Funkce pro zjištění počtu parametrů: `ParamCount`
- Funkce pro zjištění hodnoty n -tého parametru:
`ParamStr(n)`
- Hodnota parametru je vždy řetězec

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Při spuštění programu je možné zadat na příkazový řádek parametry
- Parametry jsou vhodným doplňkovým vstupem
- Obvykle používány na konfiguraci nebo úpravu činnosti programu
- Funkce pro zjištění počtu parametrů: `ParamCount`
- Funkce pro zjištění hodnoty n -tého parametru:
`ParamStr(n)`
- Hodnota parametru je vždy řetězec
- Parametry jsou číslovány od 1, hodnotu 0 má úplná cesta a název spouštěného programu

Standardní vstup
a výstup

Příkazový řádek –
parametry

Příklad získání jednoznakového
a celočíselného parametru

Proměnné prostředí

Práce se soubory

Komplexní příklad

```
procedure ComLine(var S: longint; var D: char);
  var Pom: string;
      Kod: integer;
  begin if ParamCount>0 then begin
        {je alespoň jeden parametr}
        Pom:=ParamStr(1);
        Val(Pom, S, Kod);
        if ParamCount>1 then begin
            {je i druhý parametr}
            Pom:=ParamStr(2);
            D:=Pom[1]
        end
      end
  end;
```

Proměnné prostředí

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Každý proces má prostředí, v němž lze nastavovat řetězcové proměnné

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Každý proces má prostředí, v němž lze nastavovat řetězcové proměnné
- Proměnné se dědí z prostředí nadřazeného procesu

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Každý proces má prostředí, v němž lze nastavovat řetězcové proměnné
- Proměnné se dědí z prostředí nadřazeného procesu
- Uvnitř programu lze obsah proměnné přechytit, nikoliv nastavit

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Každý proces má prostředí, v němž lze nastavovat řetězcové proměnné
- Proměnné se dědí z prostředí nadřazeného procesu
- Uvnitř programu lze obsah proměnné přečíst, nikoliv nastavit
- Existuje funkce `GetEnvironmentVariable(X)` pro získání řetězcové hodnoty proměnné `X`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Každý proces má prostředí, v němž lze nastavovat řetězcové proměnné
- Proměnné se dědí z prostředí nadřazeného procesu
- Uvnitř programu lze obsah proměnné přečíst, nikoliv nastavit
- Existuje funkce `GetEnvironmentVariable(X)` pro získání řetězcové hodnoty proměnné `X`
- Tato funkce je dostupná ze standardního modulu `SysUtils`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Každý proces má prostředí, v němž lze nastavovat řetězcové proměnné
- Proměnné se dědí z prostředí nadřazeného procesu
- Uvnitř programu lze obsah proměnné přechytit, nikoliv nastavit
- Existuje funkce `GetEnvironmentVariable(X)` pro získání řetězcové hodnoty proměnné `X`
- Tato funkce je dostupná ze standardního modulu `SysUtils`
- Analogická funkce je `GetEnv` známá i z Turbo Pascalu, dostupná v modulu `oldlinux`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Zpracování hodnoty proměnné prostředí

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Příklad

Práce se soubory

Komplexní příklad

- Zpracování hodnoty proměnné prostředí
- Jsou nastaveny dvě uživatelské proměnné NUMSTART a NUMDELIM

```
procedure Envir(var S: longint; var D: char);
  var Pom: string;
      Kod: integer;
begin Pom:=GetEnvironmentVariable('NUMSTART');
      if Pom<>' ' then Val(Pom, S, Kod);
      {proměnná neexistuje --> Pom=' '}
      Pom:=GetEnvironmentVariable('NUMDELIM');
      if Pom<>' ' then D:=Pom[1]
end;
```

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Je k dispozici procedura `Assign`, která sváže souborovou proměnnou v programu s konkrétním souborem dostupným v operačním systému

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Je k dispozici procedura `Assign`, která sváže souborovou proměnnou v programu s konkrétním souborem dostupným v operačním systému
- Konvence zápisu cesty k souboru odpovídá použitému operačnímu systému (problémy s přenositelností)

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Je k dispozici procedura `Assign`, která sváže souborovou proměnnou v programu s konkrétním souborem dostupným v operačním systému
- Konvence zápisu cesty k souboru odpovídá použitému operačnímu systému (problémy s přenositelností)
- Existence souboru není hlídána operačním systémem, není-li ošetřena v programu, dojde k běhové chybě

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Je k dispozici procedura `Assign`, která sváže souborovou proměnnou v programu s konkrétním souborem dostupným v operačním systému
- Konvence zápisu cesty k souboru odpovídá použitému operačnímu systému (problémy s přenositelností)
- Existence souboru není hlídána operačním systémem, není-li ošetřena v programu, dojde k běhové chybě
- Způsob ošetření existence souboru je závislý na operačním systému a na verzi překladače. V modulu `SysUtils` je k dispozici logická funkce `FileExists(jméno)`

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Je k dispozici procedura `Assign`, která sváže souborovou proměnnou v programu s konkrétním souborem dostupným v operačním systému
- Konvence zápisu cesty k souboru odpovídá použitému operačnímu systému (problémy s přenositelností)
- Existence souboru není hlídána operačním systémem, není-li ošetřena v programu, dojde k běhové chybě
- Způsob ošetření existence souboru je závislý na operačním systému a na verzi překladače. V modulu `SysUtils` je k dispozici logická funkce `FileExists(jméno)`
- Všechny operace se soubory byly již diskutovány (opakování)

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Konfigurace programu – čtení dvou parametrů zmíněných v předchozích příkladech

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Konfigurace programu – čtení dvou parametrů zmíněných v předchozích příkladech
- Hierarchie nastavení:

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Konfigurace programu – čtení dvou parametrů zmíněných v předchozích příkladech
- Hierarchie nastavení:
 - ① Implicitní hodnota

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Konfigurace programu – čtení dvou parametrů zmíněných v předchozích příkladech
- Hierarchie nastavení:
 - ① Implicitní hodnota
 - ② Konfigurační soubor (pokud existuje)

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Konfigurace programu – čtení dvou parametrů zmíněných v předchozích příkladech
- Hierarchie nastavení:
 - ① Implicitní hodnota
 - ② Konfigurační soubor (pokud existuje)
 - ③ Proměnná prostředí

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

- Konfigurace programu – čtení dvou parametrů zmíněných v předchozích příkladech
- Hierarchie nastavení:
 - ① Implicitní hodnota
 - ② Konfigurační soubor (pokud existuje)
 - ③ Proměnná prostředí
 - ④ Parametr z příkazového řádku

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

```
uses SysUtils; {práce s proměnnými prostředí apod.}
var Cislo, Start: longint;
    Radek: string;
    Delim: char;
procedure Konfig(var S: longint; var D: char);
    var F: text;
        R, Pom: string;
        Kod: integer;
```

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

```
begin Assign(F, ParamStr(0)+'rc');
      {test existence souboru;}
if FileExists(ParamStr(0)+'rc') then begin
      reset(F);
      readln(F, R);
      Pom:=Copy(R, Pos('=' ,R)+1, 255);
      {získání hodnoty řetězce za rovnítkem}
      Val(Pom, S, Kod);
      {převod na číselnou hodnotu}
      readln(F, R);
      Pom:=Copy(R, Pos('=' ,R)+1, 1);
      {opět hodnota za rovnítkem}
      D:=Pom[1]

end
end;
```

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

```
procedure Envir(var S: longint; var D: char);  
  var Pom: string;  
      Kod: integer;  
begin Pom:=GetEnvironmentVariable('NUMSTART');  
      if Pom<>' ' then Val(Pom, S, Kod);  
      {proměnná neexistuje --> Pom=' '}  
      Pom:=GetEnvironmentVariable('NUMDELIM');  
      if Pom<>' ' then D:=Pom[1]  
end;
```

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

```
procedure ComLine(var S: longint; var D: char);  
  var Pom: string;  
      Kod: integer;  
  begin if ParamCount>0 then begin  
    {je alespoň jeden parametr}  
    Pom:=ParamStr(1);  
    Val(Pom, S, Kod);  
    if ParamCount>1 then begin  
      {je i druhý parametr}  
      Pom:=ParamStr(2);  
      D:=Pom[1]  
    end  
  end  
end;
```

Standardní vstup
a výstup

Příkazový řádek –
parametry

Proměnné prostředí

Práce se soubory

Komplexní příklad

```
begin Start:=1; Delim:=': '; {implicitní nastavení}
Konfig(Start, Delim); {konfigurační soubor}
Envir(Start, Delim); {proměnné prostředí}
ComLine(Start, Delim); {příkazový řádek}
Cislo:=Start;
while not eof do begin
    readln(Radek);
    writeln(Cislo:5, Delim, ' ', Radek);
    Inc(Cislo)
end
end.
```