PB152cv/07 Cvičenie 7

(nahradzovanie programov v procesoch, správa prostredia)

Jozef Sabo

10. 11. 2025

Masarykova Univerzita, Brno

Cvičenie 07

Duplikácia popisovača súboru s výberom nového popisovača

```
int dup2(int filedes, int filedes2);
```

`int filedes` – popisovač súboru, pre ktorý chceme nové číslo popisovača

`int filedes2` – nové, požadované číslo popisovača súboru

slúži na zduplikovanie popisovača, ak napr chceme nahradiť súbory pre vstup/výstup inými súbormi (napr. rúrou)

vtedy existujúce popisovače (makro `stdin` pre štandardný vstup) nahradíme novými popisovačmi – `dup2(pipe_fd, stdin);`

pozor! – pôvodný popisovač (odkazovaný `filedes2`) je uzavretý, ak existuje

Nahradzovanie programov v procesoch

```
int exec[vl][$ep](...);
```

volanie nahradí program v rámci procesu (nevytvára proces nový)

v – vkladanie premenných poľom – do volaní sa vkladajú polia argv a envp, ktorých posledný prvok musí byť NULL

l – variadické vkladanie premenných – nekonštruujú sa polia pre argv a prípadne envp, ale jednotlivé hodnoty sa vkladajú priamo do volania funkcie a ukončujú sa nulovým smerníkom NULL

e – vkladáme premenné prostredia, v prípade funkcie bez, berie sa aktuálne prostredie

p – namiesto cesty sa vkladá iba názov príkazu, ktorý sa vyhľadáva v premennej prostredia $PATH

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

napr.

```
char *argv[] = { „arg1“, „arg2“, NULL };
execv( „/bin/sh“, argv );
```

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

napr.

```
char *argv[] = { „arg1“, „arg2“, NULL };
char *envp[] = { „PATH=ABC“, „PASSWORD=pass“, NULL };
execve( „/bin/sh“, argv, envp );
```

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

napr.

```
char *argv[] = { „arg1“, „arg2“, NULL };
execvp( „sh“, argv );
```

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

napr.

```
execl( „/bin/sh“, „arg1“, „arg2“, NULL );
```

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

napr.

```
char *envp[] = { „PATH=ABC“, „PASSWORD=pass“, NULL };
execle( „/bin/sh“, „arg1“, „arg2“, NULL, envp );
```

Nahradzovanie programov v procesoch

```
int execv(const char *path, char *const argv[]);
int execve(const char *path, char *const argv[], char *const envp[]);
int execvp(const char *file, char *const argv[]);

int execl(const char *path, const char *arg0, ... /*, (char *)0 */);
int execle(const char *path, const char *arg0, ... /*, (char *)0, char
*const envp[]*/);
int execlp(const char *file, const char *arg0, ... /*, (char *)0 */);
```

napr.

```
execlp( „sh“, „arg1“, „arg2“, NULL );
```