PB152cv/07 Cvičenie 5

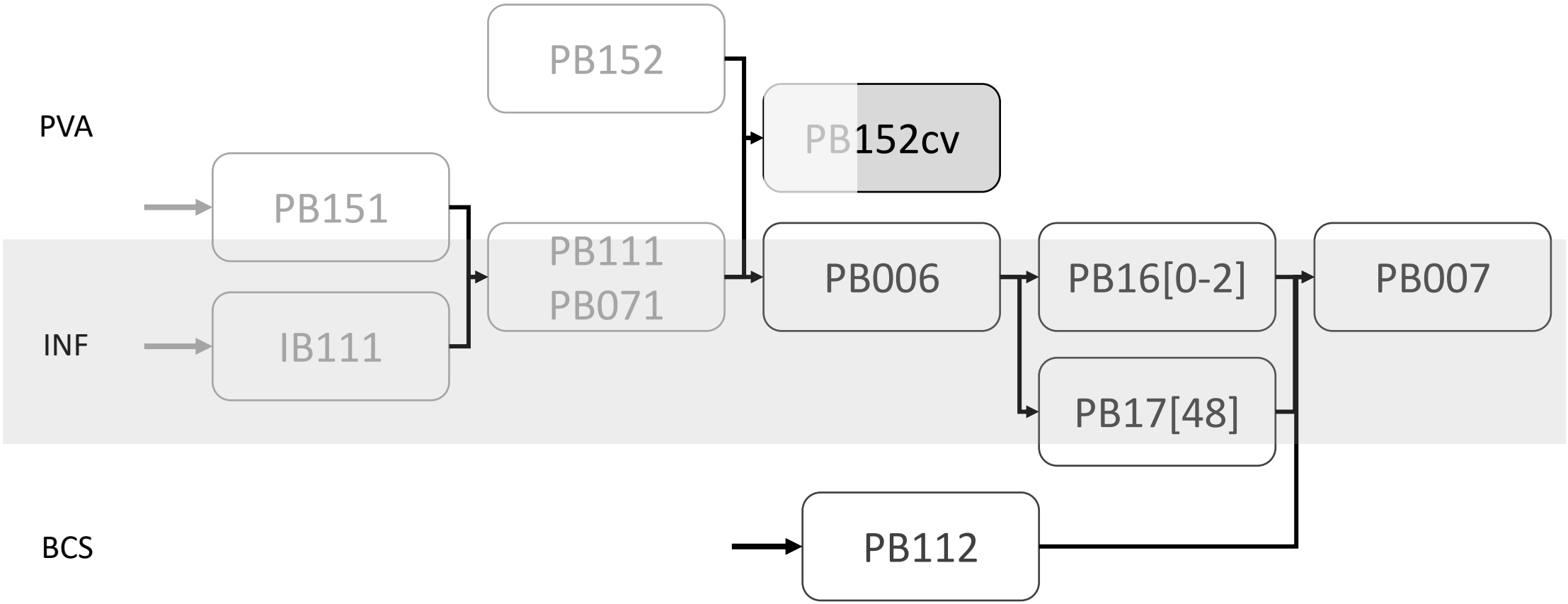
(Práca s adresármi, zisťovanie údajov o súboroch, premenovávanie)

Jozef Sabo

27. 10. 2025

Masarykova Univerzita, Brno

Kde sme


PB152
PVA
PB152cv
PB151
PB111
PB071
PB006
PB16[0-2]
PB007
INF
IB111
PB17[48]
BCS
PB112

Cvičenie 05

Otváranie súboru

```
int openat(int fd, const char *path, int oflag, mode_t mode);
int open(const char *path, int oflag, mode_t mode);
```

`int fd` – popisovač súboru reprezentujúci adresár, v ktorej chceme súbor otvoriť
pre aktuálny adresár makro `FD_ATCWD`

`const char *path` – názov súboru

`int oflag` – určujeme ním typ prístupu k súboru

`O_RDONLY, O_WRONLY, O_RDWR, O_APPEND` – v bežnom zmysle

`O_CREAT` – vytvorí obyčajný súbor ak ešte neexistuje

`O_DIRECTORY` – otvára sa tým iba adresár, nie iný druh súboru

`O_CLOEXEC` – po vykonaní programu (exec) sa popisovač uzavrie

`mode_t mode` – dobrovoľný parameter pre mód (chmod) súboru (prístup ogp) – napr. 0666

Zatváranie súboru

```
int close(int fd);
```

`int fd` – popisovač súboru, ktorý nám vrátilo volanie `openat()`/`open()`

Zisťovanie informácii o i-uzle (súbore v súborovom systéme)

```
int fstatat(int fd, const char *path, struct stat *buf, int flag);
```

`int fd` – popisovač súboru reprezentujúci adresár, kde sa nachádza súbor

`const char *path` – názov súboru

`struct stat *buf` – štruktúra, v ktorej budú volaním uložené informácie o súbore

`int oflag` – určujeme ním typ prístupu k súboru (zvyčajne `0`)

`AT_SYMLINK_NOFOLLOW` – ak by mal byť odkazovaný i-uzol mäkký odkaz, vráti informácie o odkaze samotnom, nie súbore ním odkazovanom

Štruktúra `struct stat` (nie vyčerpávajúca, viac info v `sys/stat.h`)

```
struct stat {
  dev_t      st_dev;      /* ID of device containing file */
  ino_t      st_ino;      /* Inode number */
  mode_t     st_mode;     /* File type and mode */
  nlink_t    st_nlink;    /* Number of hard links */
  uid_t      st_uid;      /* User ID of owner */
  gid_t      st_gid;      /* Group ID of owner */
  dev_t      st_rdev;     /* Device ID (if special file) */
  off_t      st_size;     /* Total size, in bytes */
  blksize_t  st_blksize;  /* Block size for filesystem I/O */
  blkcnt_t   st_blocks;   /* Number of 512 B blocks allocated */

  struct timespec  st_atim;  /* Time of last access */
  struct timespec  st_mtim;  /* Time of last modification */
  struct timespec  st_ctim;  /* Time of last status change */
}
```

Typ súboru a prístupové práva (`mode_t st_mode`, info v `sys/stat.h`)

`S_ISDIR(m)` – `true`, ak v `st_mode` je adresár

`S_ISREG(m)` – `true`, ak v `st_mode` je obyčajný súbor

`S_ISLNK(m)` – `true`, ak v `st_mode` je mäkký odkaz

`S_I<MODE><WHO>` – masky pre jednotlivé vlastnosti

`MODE`s – `RWX, R, W, X`

`WHO`s – `U, USR, G, GRP, O, OTH`

napríklad:

`S_IRGRP` – vlastnícka skupina môže čítať súbor

`S_IRWXO` – každý nevlastník môže čítať, zapisovať, spúšťať

Duplikácia popisovača súboru

```
int dup(int fd);
```

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

slúži na zduplikovanie popisovača ak napr. pôvodný nepatrí volajúcemu (vytvorí nové číslo popisovača)

ku novému popisovaču sa správame rovnako, ako keby sme ho otvorili pomocou `open()`

pozor! – popisovač na súbor iba odkazuje, zmeny pomocou jedného popisovača (napr. posun v súbore) sa prejavia aj na druhom popisovači

Otváranie adresára (systémovo nešpecificky)

`DIR *fdopendir(int fd);` – otvorenie adresára systémovo nešpecifickým rozhraním, vracia smerník na štruktúru `DIR` alebo `NULL`

`int fd` – popisovač súboru reprezentujúci adresár, ktorý chceme otvoriť cez tohto rozhranie

`void rewindir(DIR *dirp);` – presun na začiatok adresára, pozor! – toto volanie nevie zlyhať

`DIR *dirp` – smerník na štruktúru `DIR` vrátenú `fdopendir()`

Čítanie z adresára (systémovo nešpecificky)

```
struct dirent *readdir(DIR *dirp);
```

`DIR *dirp` – smerník na štruktúru `DIR` vrátenú `fdopendir()`

```
struct dirent {
  ino_t      d_ino;     /* Inode number */
  char[]     d_name;    /* File name */
}
```

pozor! – medzi dvoma volaniami `readdir()` sa môže pamäť reprezentovaná návratovou hodnotou zneplatniť, preto nemôžeme spoliehať na to, že uložené dáta ostanú na svojom mieste

Zatváranie adresára (systémovo nešpecificky)

```
int closedir(DIR *dirp);
```

`DIR *dirp` – smerník na štruktúru `DIR` vrátenú `fdopendir()`

pozor! – funkcia `closedir()` zatvára aj popisovač súboru ňou reprezentovaný

Presunutie súboru

```
int renameat(int oldfd, const char *old, int newfd, const char *new);
```

`int oldfd` – popisovač súboru reprezentujúci adresár, kde sa nachádza súbor aktuálne

`const char *old` – aktuálny názov súboru

`int newfd` – popisovač súboru reprezentujúci adresár, kam sa súbor presunie

`const char *new` – nový, požadovaný, názov súboru