PB152cv/07 Cvičenie 9

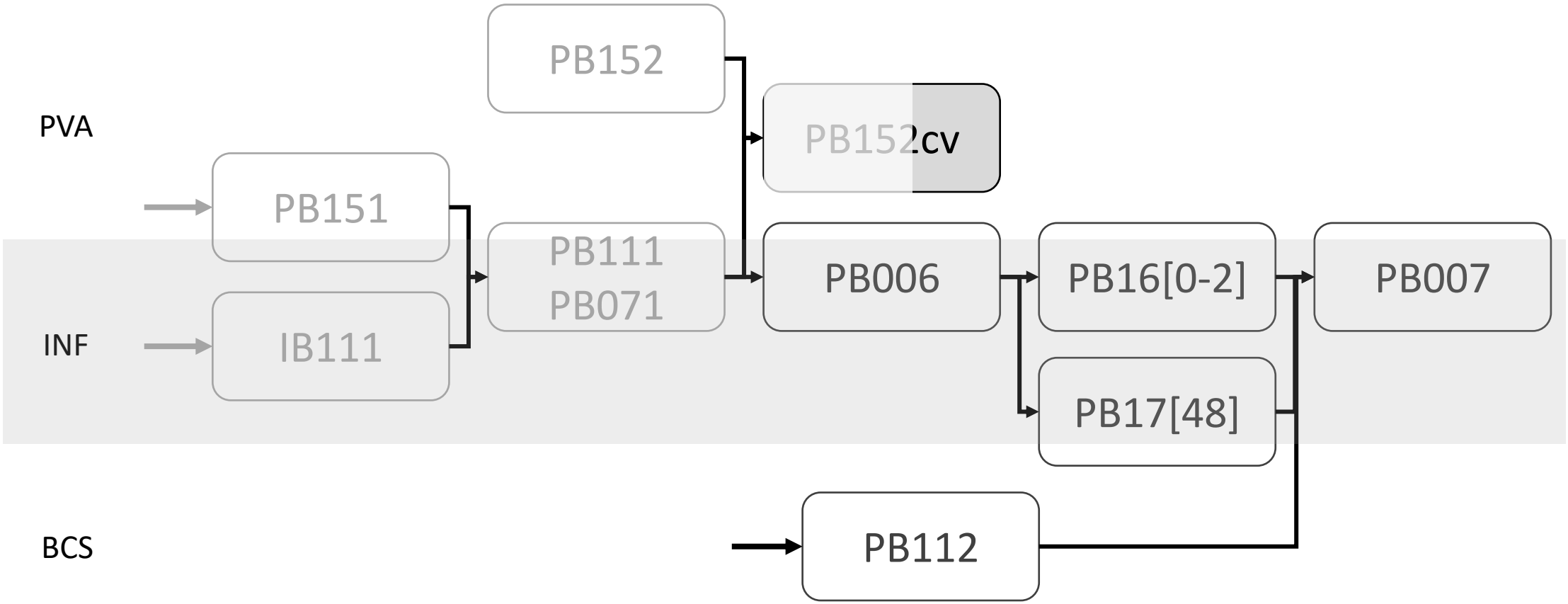
(Rozdeľovanie procesov (vidličkovanie), medziprocesová komunikácia)

Jozef Sabo

24. 11. 2025

Masarykova Univerzita, Brno

Kde sme


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

Cvičenie 09

Rozdeľovanie procesov – vytváranie kópie procesu

```
pid_t fork();
```

volanie vytvorí nový proces (dieťa) ako kópiu aktuálneho (rodič)

v rodičovskom procese je návratovou hodnotou Proces IDentifier (PID) detského procesu (alebo samozrejme `-1` pri chybe)

v detskom procese je návratovou hodnotou `0`

adresný priestor oboch procesov je rovnaký – tzn. majú oba prístup k rovnakým dátam

kopírujú sa aj popisovače otvorených súborov, tie majú vzťah ako pri použití `dup()` (a to najmä že ich treba oba zatvoriť a že zdieľajú rovnakú pozíciu v súbore)

pozor! – v detskom procese sa dáta kopírujú až po pokuse o zápis (copy on write – skriptá PB152 z roku 2025, strana 10, odrážka 1.3.4)

Ukončenie procesu (dieťa)

```
void exit( int status );
```

`int status` – návratová hodnota (aktuálneho procesu) – iba spodných 8 bitov tejto hodnoty je odovzdaných rodičovskému procesu

ukončenie vlastného procesu (samého seba)

zvyčajne sa používa v detskom procese

Čakanie na ukončenie procesu (rodič)

```
pid_t waitpid( pid_t pid, int *stat_loc, int options );
```

`pid_t pid` – PID detského procesu, na ktorý sa čaká (alebo iné hodnoty v `man 3p wait`)

`int *stat_loc` – smerník na typ `int` kam sa uloží návratová hodnota (odovzdaná volaním `exit()` v detskom procese)

`int options` – príznaky, používame hlavne

- `0` – blokuje, dokiaľ nie je detský proces ukončený
- `WNOHANG` – neblokuje, návratová hodnota môže byť `-1` a zároveň `errno EAGAIN`

hodnota `pid` nemôže byť systémom pridelená inému procesu, pokiaľ na nej neprebehlo systémové volanie `waitpid()` (alebo nepriamo volaním `wait()`)

Čakanie na ukončenie procesu (rodič) – pomocné makrá

pre pohodlné spracovanie výsledku systémového volania `waitpid()` používame makrá

`WIFEXITED( stat_loc )` – vyhodnotí sa na nenulovú hodnotu ak proces skončil v poriadku

`WEXITSTATUS( stat_loc )` – vyhodnotí sa na návratovú hodnotu odovzdanú cez exit() v detskom procese (spodných 8 bitov)

`WIFSIGNALED( stat_loc )` – nepoužívame

`WTERMSIG ( stat_loc )` – nepoužívame

`WIFSTOPPED( stat_loc )` – nepoužívame

`WSTOPSIG( stat_loc )` – nepoužívame

`WIFCONTINUED( stat_loc )` – nepoužívame

Rúry

```
int pipe( int filedes[ 2 ] );
```

`int filedes[ 2 ]` – pole dvoch prvkov typu `int`, kam sa zapíše popisovač pre čítanie a zápis

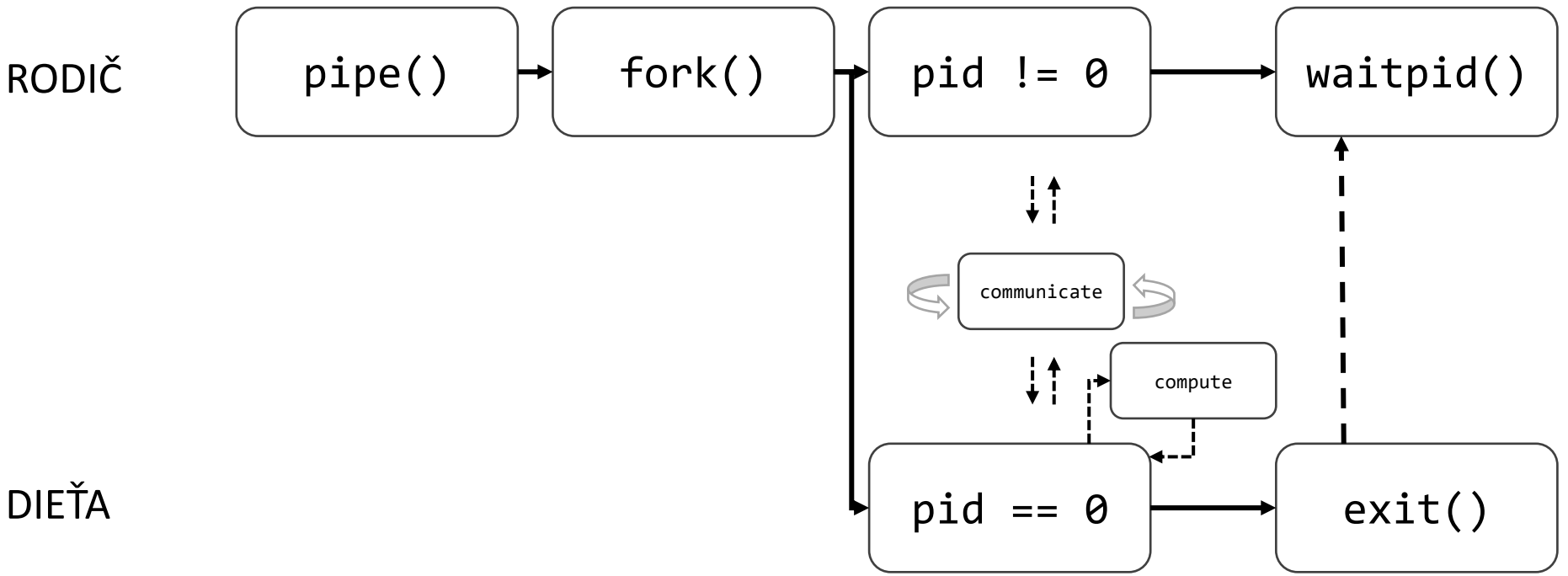
po úspešnom volaní sa vytvorí prepojená rúra, kde `filedes[ 0 ]` obsahuje popisovač pre čítanie z tejto rúry a `filedes[ 1 ]` obsahuje popisovač pre zápis do tejto rúry

čítanie a zápis sa vykonáva bežným `read()` a `write()`

v rúre sa nedá pohybovať (pomocou `lseek()`) a raz prečítané dáta sa nedajú prečítať znovu

ukončenie zápisu/prijímania sa oznámi druhej strane zatvorením vlastného popisovača (pomocou `close()`)

Príklad výpočtu



Príklad výpočtu

RODIČ

pipe()

DIEŤA

```
int parent_to_child[ 2 ] = pipe();
int child_to_parent[ 2 ] = pipe();

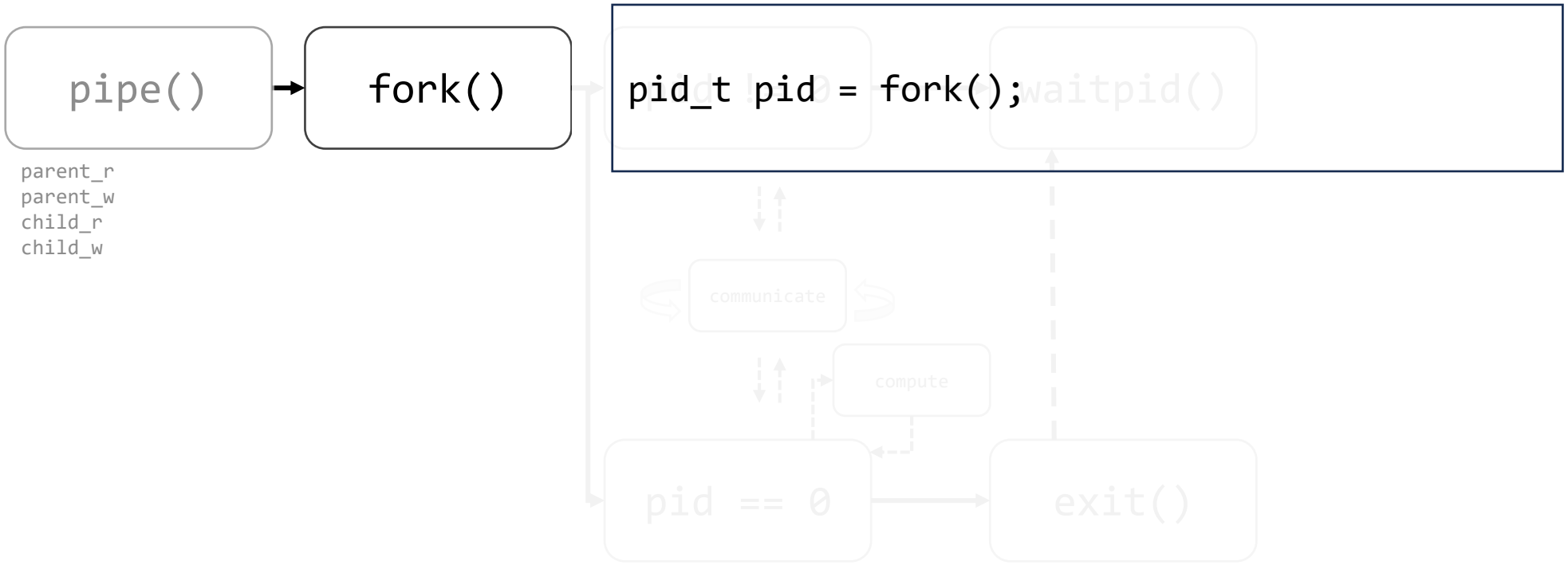
int parent_r = child_to_parent[0];
int parent_w = parent_to_child[1];

int child_r = parent_to_child [0];
int child_w = child_to_parent[1];

// creating two pipes for full duplex
communication
```

Príklad výpočtu

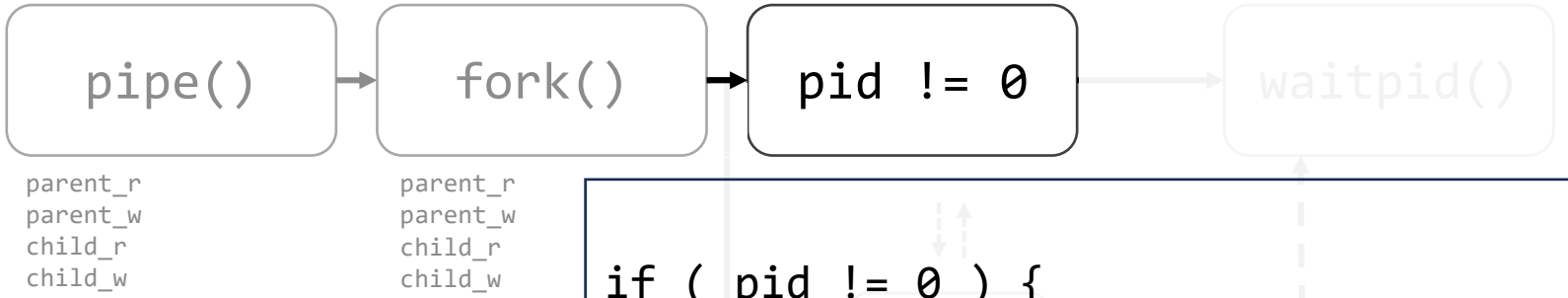
RODIČ



DIEŤA

Príklad výpočtu

RODIČ



DIEŤA

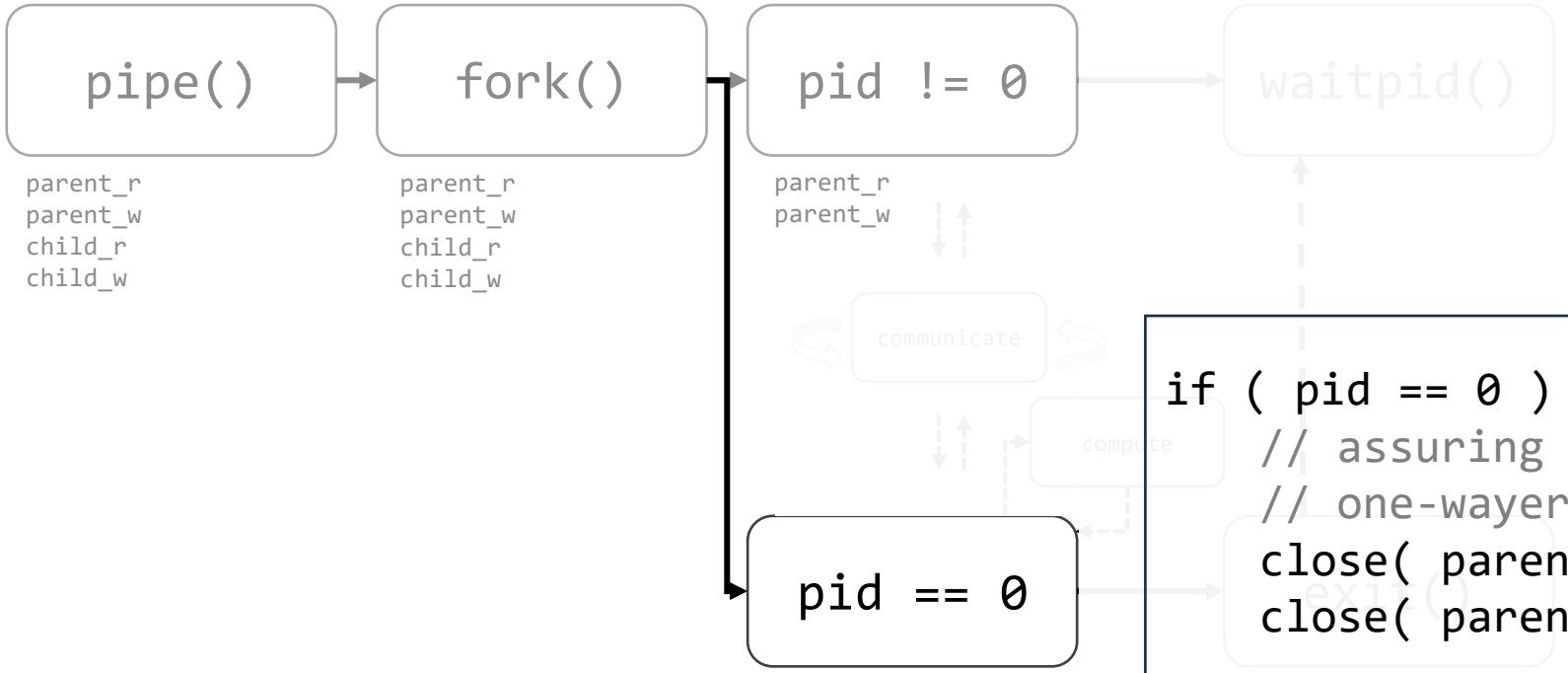
```
if ( pid != 0 ) {
    // assuring the pipe to be one-wayer
    // assures also that close() on the oposite
    // side of the pipe finishes the communication
    close( child_r );
    close( child_w );

    ...
}
```

Príklad výpočtu

RODIČ

DIEŤA



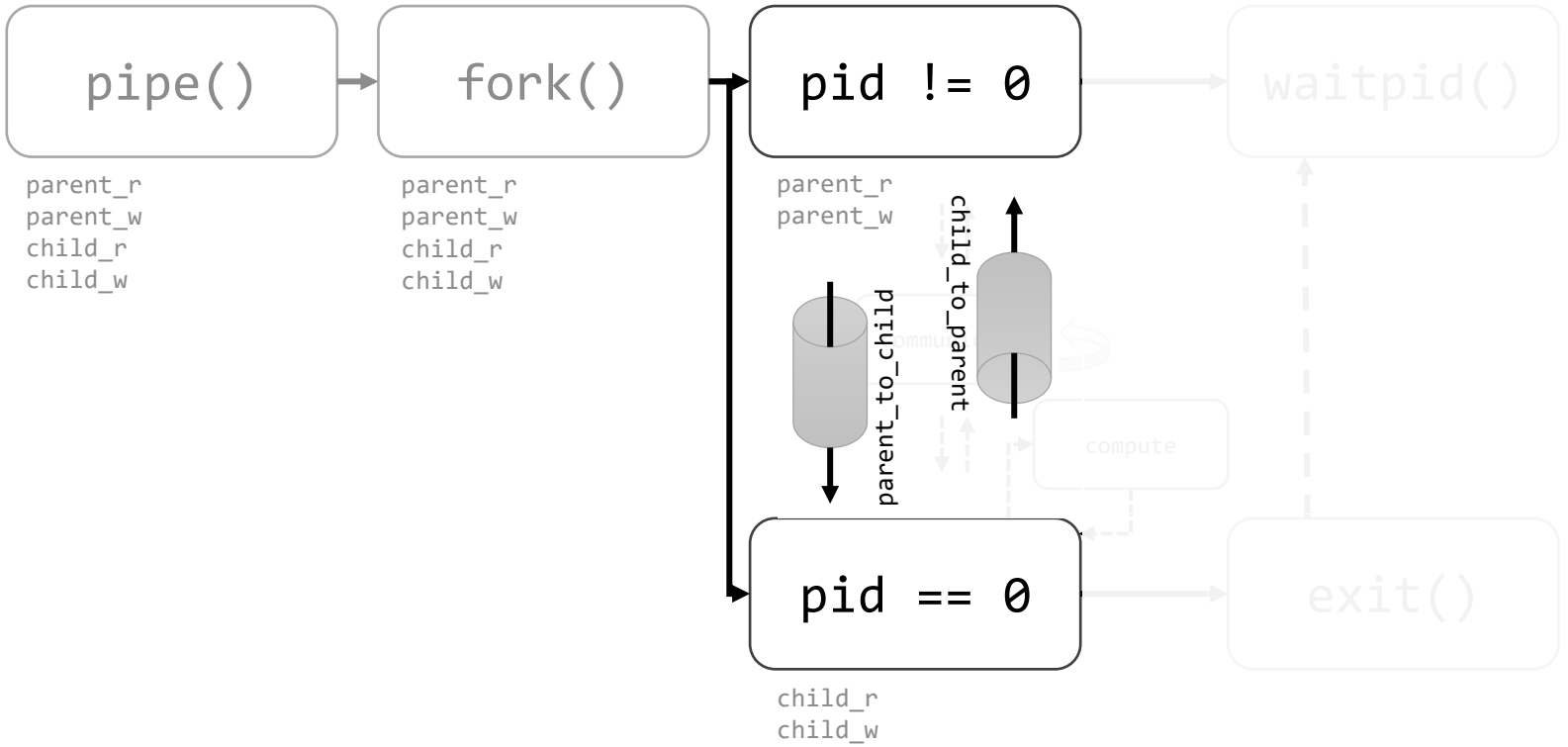

```
if ( pid == 0 ) {
    // assuring the pipe to be
    // one-wayer
    close( parent_r );
    close( parent_w );

    ...
}
```

Príklad výpočtu

RODIČ

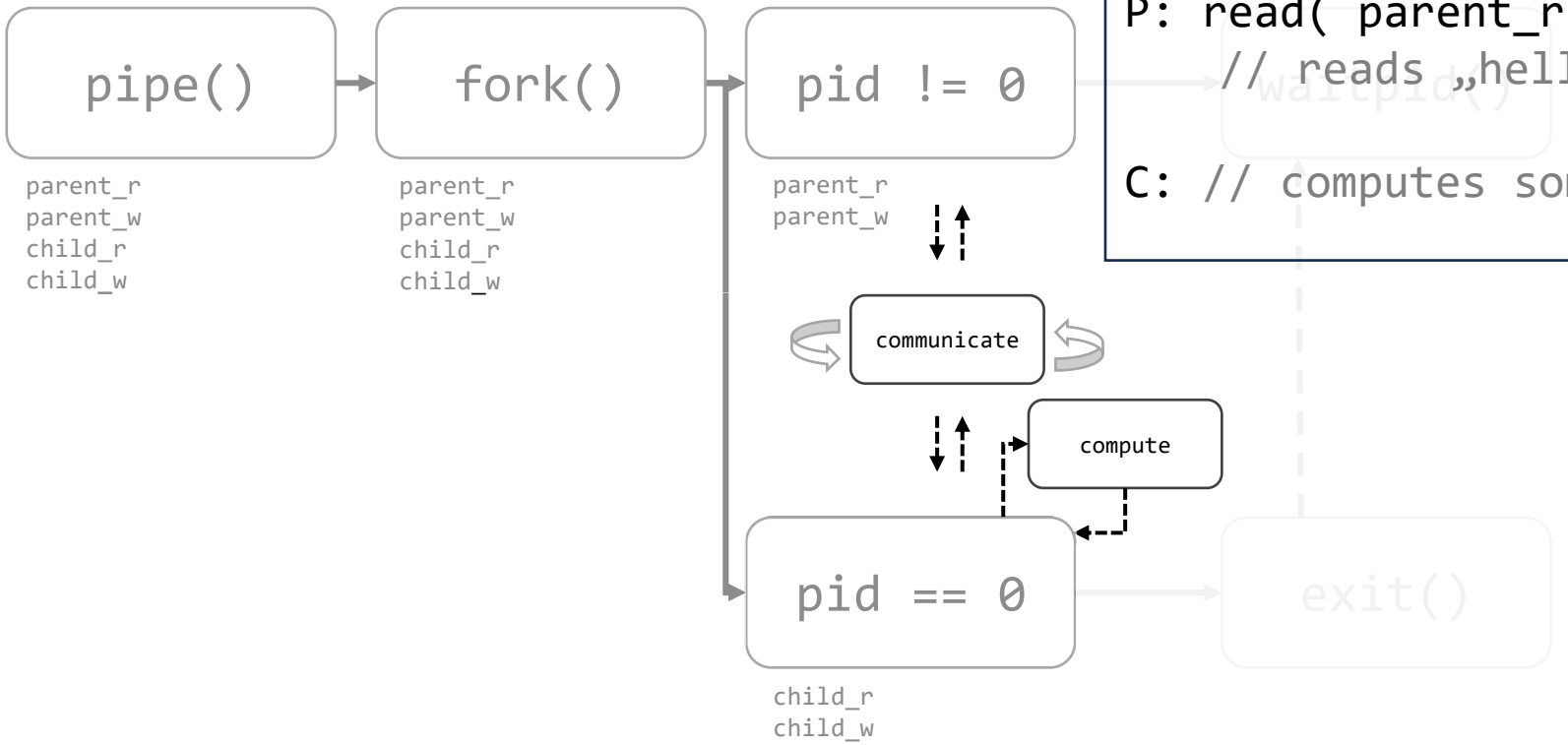
DIEŤA



Príklad výpočtu

RODIČ

DIEŤA




```
P: write( parent_w, "hi", ... );
C: read( child_r, ... );
    // reads „hi“

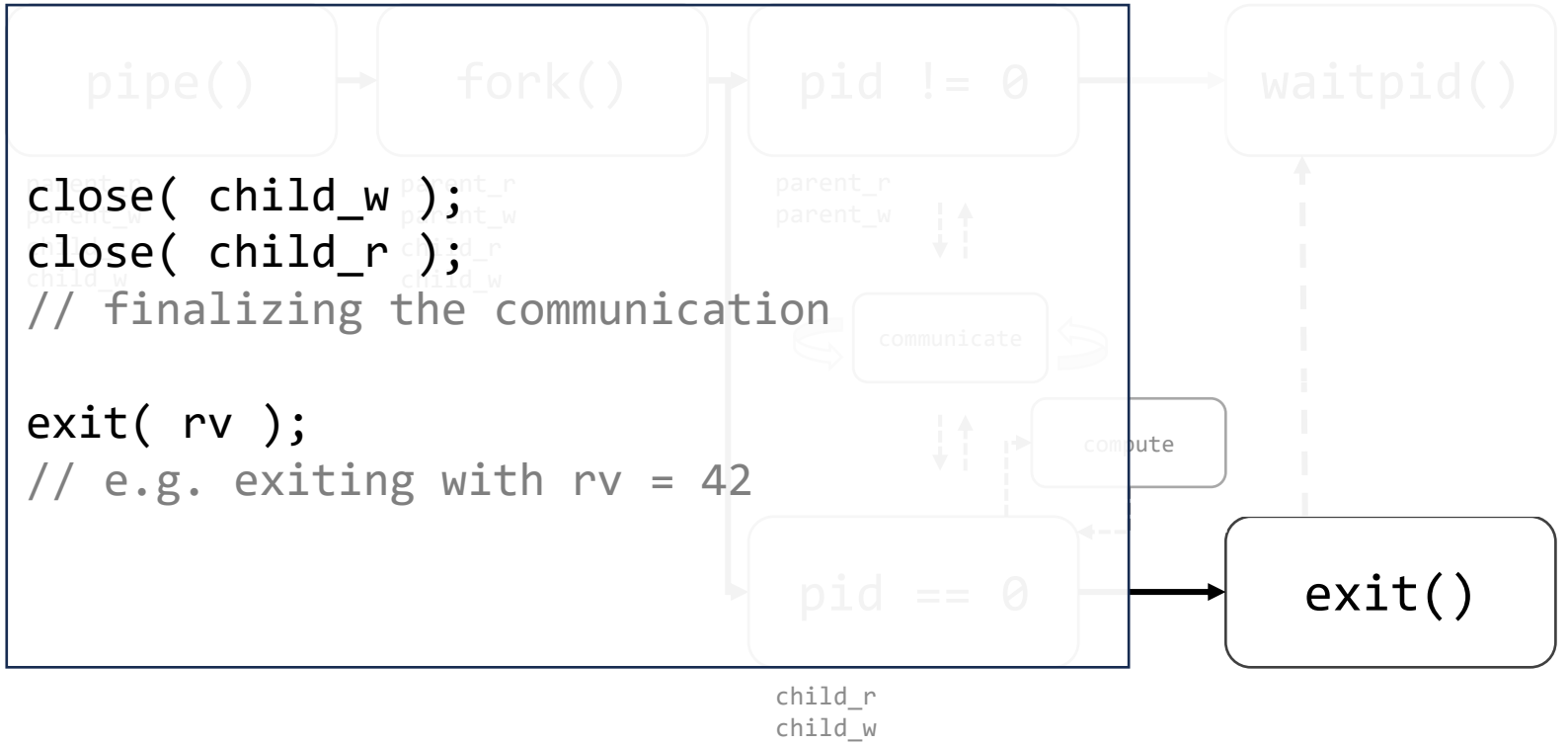
C: write( child_w, „hello", ... );
P: read( parent_r, ... );
    // reads „hello“

C: // computes something
```

Príklad výpočtu

RODIČ

DIEŤA



Príklad výpočtu

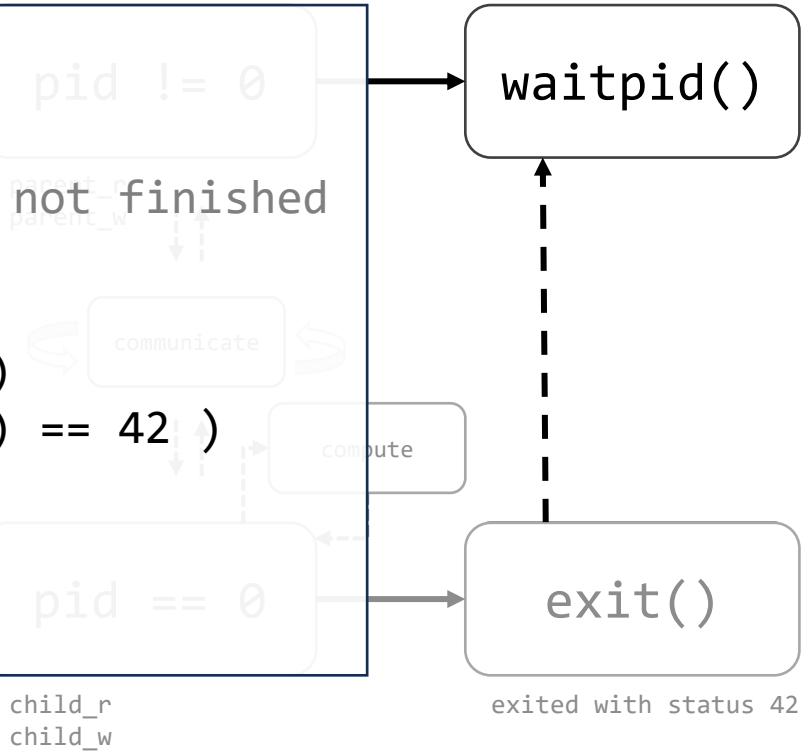
RODIČ

DIEŤA

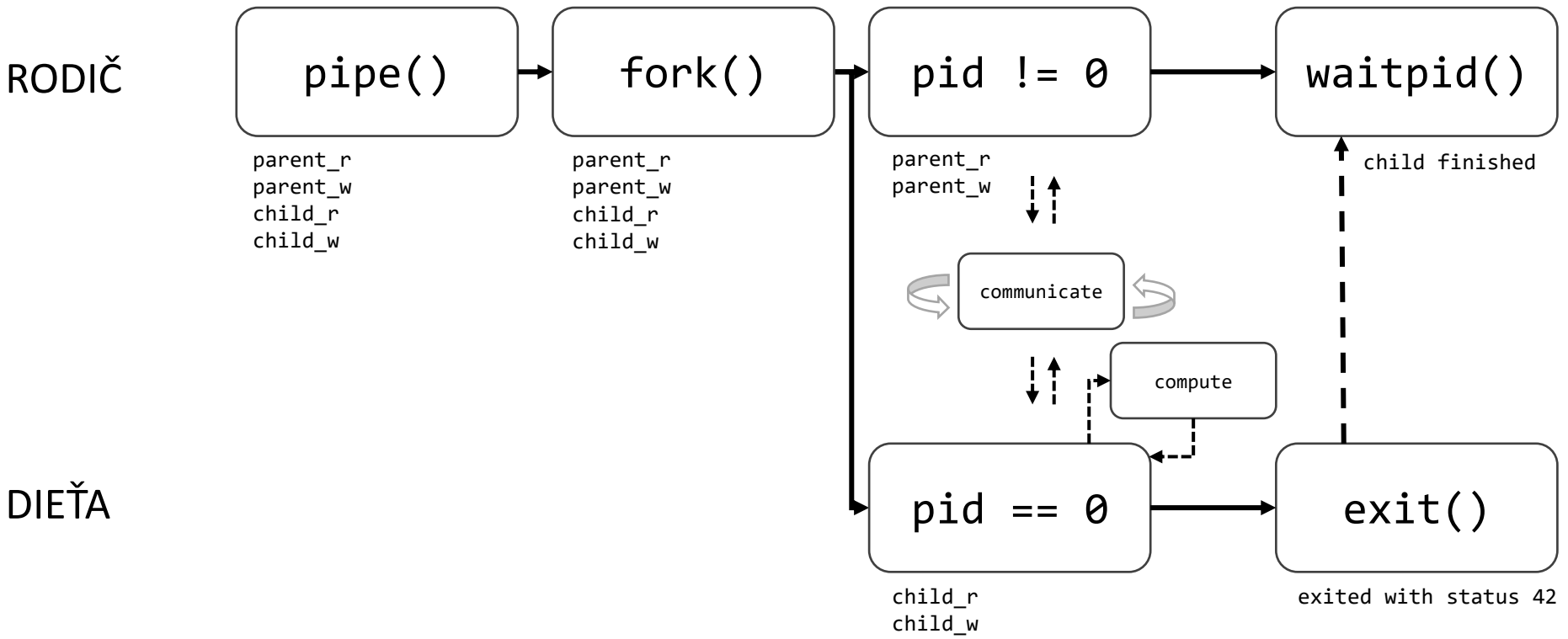
```
int status;

// going blocking till child not finished
waitpid( pid, &status, 0 );

if (    WIFEXITED(   status )
     && WEXITSTATUS( status ) == 42 )
    // all good
else
    // ja pierdolę
```



Príklad výpočtu



Príklad výpočtu

