



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale
2007-2013



Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Sisteme de operare

11. Lucrul cu procese în Linux

Operații cu procese

- Crearea proceselor
- Terminarea proceselor
 - voluntară (exit)
 - involuntară (excepții, erori fatale)
- Întreruperea proceselor
 - semnale
- Comunicare interproces
- Sincronizare

Crearea unui proces

- Execuția unui program; rularea unei comenzi
- Un proces se creează dintr-un proces existent
- Relație părinte-copil
- Cine este procesul părinte în cazul execuției unei comenzi (externe)?
- Executabilul este imaginea noului proces
- Procesului i se asociază resurse
- Devine o entitate planificabilă pe procesor

Crearea unui proces în Unix

```
pid_t pid;

pid = fork();
switch (pid) {
    case -1: /* fork failed */
        perror("fork");
        exit(EXIT_FAILURE);
    case 0: /* procesul copil */
        ...
    default: /* procesul părinte */
        printf("Created process with pid %d\n", pid);
    ...
}
```

fork

- Se apelează o dată dar se întoarce de două ori
 - în procesul copil întoarce 0
 - în procesul părinte întoarce pid-ul procesului copil
 - de ce?
- Procesul copil este o copie a procesului părinte (spațiu de adresă)
- Resursele procesului părinte sunt partajate în procesul copil (fișiere, socketi)

Execuția unui proces în Unix

- Pentru execuția unui program se înlocuiește imaginea curentă de executabil cu una nouă – apelurile din familia exec
- Se creează un spațiu nou de adrese
- Se încarcă executabilul
- PID-ul nu se schimbă
- Descriptorii de fișier sunt folosiți în continuare (doar dacă nu au flag-ul `CLOSE_ON_EXEC`)

Încheierea execuției

- Când se termină un proces?
 - s-a încheiat programul (return exit_code din main)
 - se programează terminarea procesului (exit)
 - alt program termină procesul curent (kill)
- Încheierea forțată
 - semnale
 - abort – SIGABRT

Așteptarea terminării unui proces

- Procesul apelant este blocat până la terminarea procesului specificat (proces copil)
 - `wait`, `waitpid`, `WIFEXITED`, `WIFSIGNALED`
- Cum sunt implementați operatorii shell de mai jos?
 - `&`
 - `;`
 - `||`
 - `&&`