

LAB 1

fork-exec-wait

INF01142 - Sistemas Operacionais IN

Prof.: Alexandre Carissimi

Monitor: Eduardo Roloff (mestrando)

Roteiro

- Ferramental
- Revisão
- Makefile
- Programa exemplo
- Exercício

Ferramental

- Todo estudante de Computação precisa:
 - Ter um bom editor de texto
 - Saber usar o “man”
 - Conhecer uma linguagem de programação razoável
 - Isso significa C
 - Google is your friend

Revisão

- **fork()**
 - Cria um processo filho igual ao pai
 - Mas com próprio PID no SO
- **wait()**
 - Pai bloqueia e espera o final da execução de algum de seus filhos (qualquer um)
 - **Dica:** `waitpid()`

Revisão

- Família **exec()**
 - Inicia um outro programa

Makefile

- Arquivo usado para compilação de projetos
 - Pense em centenas de fontes diferentes
- Evita recompilar todo o projeto
 - Compila só o que foi modificado
- Sintaxe: `make`
- Exemplo no arquivo baixado do moodle

Exemplo

```
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>

int main( ) {
    pid_t pid;
    int status;

    if ((pid = fork()) != 0) {
        printf("Processo pai esperando filho PID=%d\n", pid);
        wait(&status);
        printf("Filho terminou com o status: %d\n", WEXITSTATUS(status));
        exit(0);
    }
    else {
        printf("Filho: troca de codigo executavel...\n");
        execl("/bin/ls", "ls", "-la", NULL);
        printf("--->Soh³ volta aqui se deu erro na chamada exec!!\n");
    }
}
```

Exercício 1

- 1) Abra o arquivo `fork.c` e analise o código
 - Identifique o que o código está fazendo
 - Compile e execute

- 2) Remova os comentários do código
 - Compile e execute novamente
 - Explique o que mudou na segunda execução
 - **Dica:** Não são os `printf` extras

Exercício 2

- Crie um programa para cálculo de fatorial
 - Recursivo
 - Ao invés de executar a recursão, crie um novo processo para efetuar o cálculo
 - Some os resultados até chegar ao pai novamente
 - Somente o pai deve exibir o resultado