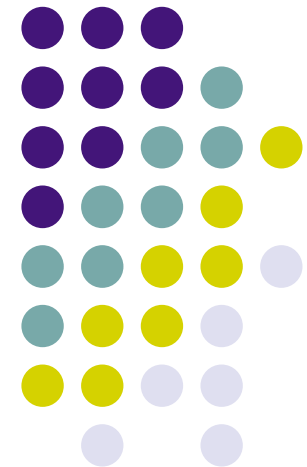
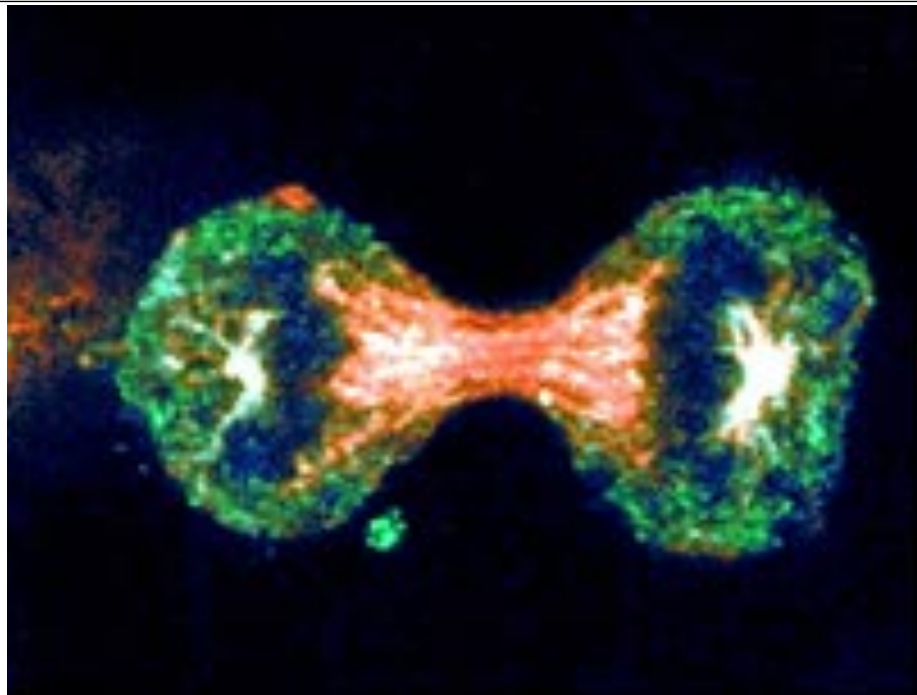


# Criação de Processos

**Fork( )**  
**Exec( )**



# Chamadas de Sistema: Process Management



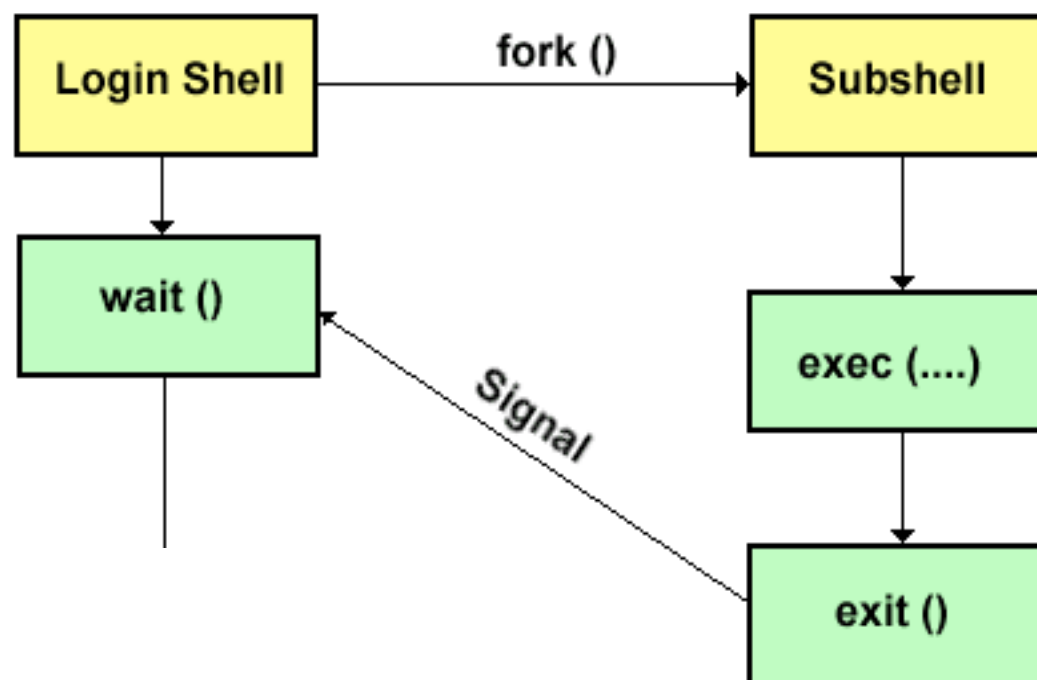
## Gerenciamento de processos

Chamada	Descrição
<code>pid = fork( )</code>	Crie um processo filho idêntico ao processo pai
<code>pid = waitpid(pid, &amp;statloc, options)</code>	Aguarde um processo filho terminar
<code>s = execve(name, argv, environp)</code>	Substitua o espaço de endereçamento do processo
<code>exit(status)</code>	Termine a execução do processo e retorne o estado

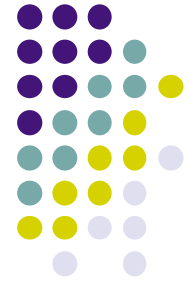
Dica: Testar sempre como o filho terminou

```
waitpid(-1, &status, 0);  
if (! WIFEXITED(status) ) print("O processo filho  
nao terminou corretamente!\n");
```

# Chamada fork( ) / exec( )



# Esboço de uma shell



```
while (TRUE) {  
    type_prompt( );  
    read_command (command, parameters)  
  
    if (fork() != 0) {  
        /* Parent code */  
        waitpid( -1, &status, 0);  
    } else {  
        /* Child code */  
        execve (command, parameters, 0);  
    }  
}
```

/\* repeat forever \*/  
/\* display prompt \*/  
/\* input from terminal \*/  
  
/\* fork off child process  
  
/\* wait for child to exit \*/  
  
/\* execute command \*/

# Variáveis de Ambiente



São um conjunto de valores dinâmicos nomeados que podem afetar a forma como os processos em execução se comportarão.

Cada processo tem o seu próprio conjunto de variáveis de ambiente.

Quando um processo é criado ele herda um ambiente duplicado de seu processo pai, exceto por alterações explícitas feitas pelo pai quando ele cria o filho.

Os programas em execução podem acessar os valores das variáveis de ambiente para fins de configuração. Exemplos de variáveis de ambiente incluem:

- **path**- lista diretórios que o shell procura, para os comandos que o usuário pode digitar sem ter que fornecer o caminho completo.
  - **temp** - local onde os processos podem armazenar arquivos temporários
  - **UserProfile** - indica onde o diretório home do usuário está localizado no sistema de arquivos.
  - **AppData** - para armazenar as configurações/settings do aplicativo.
- Temp - local onde os processos podem armazenar arquivos temporários.

# Variações de exec()



```
#include <unistd.h>  
extern char **environ;
```

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

**Obs:** argv[] e envp[] precisam terminar com NULL  
O 1o. elemento de argv[] deve ser o nome/path do executável.

# Perguntas?

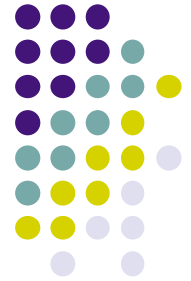


# Exercícios



- 1) Faça um programa para criar dois processos, o pai escreve seu pid e espera o filho terminar e o filho escreve o seu pid e termina.
- 2) Agora, usando a mesma estrutura de processos pai e filho, declare uma variável visível ao pai e ao filho, no pai inicialize a variável com 1 e imprima seu valor antes do **fork()**. No filho, altere o valor da variável para 5 e imprima o seu valor antes do **exit()**. Agora, no pai, imprima novamente o valor da variável após o filho ter alterado a variável - após a **waitpid()**. Justifique os resultados obtidos.
- 3) Use o programa anterior para ordenar um vetor de 10 posições. O filho ordena o vetor e o pai exibe os dados do vetor antes do **fork()** e depois do **waitpid()**. Eles usarão o mesmo vetor na memória? Justifique.
- 4) Modifique o programa anterior para que o filho execute um programa elaborado por você, que mande imprimir uma mensagem qualquer no vídeo, por exemplo, “alo mundo”. Em seguida altere o programa do item 4 para o filho executar o programa echo da shell.





# Entregáveis

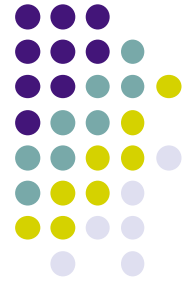
Um relatório (em um único arquivo de texto, formato ASCII) contendo, para cada exercício resolvido:

- Código fonte do programa(s)
- A chamada do programa e sua saída
- Uma explicação sobre a razão de ter obtido esse resultado

Enviar o relatório para

[fbeneditovm@gmail.com](mailto:fbeneditovm@gmail.com) e [endler@inf.puc-rio.br](mailto:endler@inf.puc-rio.br) até a 4a.feita às 23:59

# URL das aulas práticas



- [www.inf.puc-rio.br/~endler/courses/inf1019/transp/aulas-praticas/](http://www.inf.puc-rio.br/~endler/courses/inf1019/transp/aulas-praticas/)