

Processos Pai e filho

↓
processo criador

↳ processo criado

- filhos podem criar novos processos
- partilha de Recursos
 - pai e filhos partilham Recursos
 - filhos partilham uma parte dos Recursos do pai
 - pai e filhos não partilham Recursos
- execução
 - pai e filhos executam em paralelo
 - pai espera que filho(s) terminem

fork() - chamada ao sistema

- cria um processo que consiste na cópia do espaço de endereçamento do processo pai
- Ambos os processos continuam a execução após o fork
- dependendo do SO, o fork duplica todas as threads ou só aquela em que foi executado

exec() - chamada ao sistema

- substitui o espaço de memória do processo ou do novo processo (faz load de um ficheiro binário para a memória)



Primitivas de sincronização

dos semáforos - up() e down()

- Os semáforos implicam um valor inteiro que, a partir da sua inicialização, é alterado apenas por 2 operações - up e down

- alteram sempre o valor do semáforo, podendo bloquear ou desbloquear o processo ou não ter qualquer ação - depende se o valor é $>$ ou $<$ que zero

das variáveis de condição - signal() e wait()

- existem no contexto dos monitores
 - x .wait() \rightarrow bloqueia o processo que o invoca e liberta o monitor
 - x .signal() \rightarrow acorda um dos processos (se existir) bloqueando nesta variável. Se não existir, nada acontece.
- \rightarrow O signal contrasta com o up na medida que nos semáforos é sempre alterado mas no signal pode não ser

Condições necessárias para a existência de deadlock

- exclusão mútua

↳ cada recurso de está livre ou foi atribuído a um só processo

- espera com retenção

↳ cada processo, ao requerir um novo recurso, mantém na sua posse os recursos anteriormente solicitados

- não libertação

↳ nenhum, a não ser o próprio processo, pode decidir sobre a libertação de um recurso que lhe tenha sido atribuído

- espera circular

↳ formar-se uma cadeia circular de processos em que cada processo requer 1 recurso que lhe tenha sido atribuído

Objetivos da memória virtual

- eficiência

- segurança

↳ impedir a atribuição de conteúdo de memória de outros processos

- transparência

↳ acesso a muita memória como se toda a memória lhe pertencesse

- partilha de memória

↳ vários processos acedem à mesma zona de memória de forma controlada

