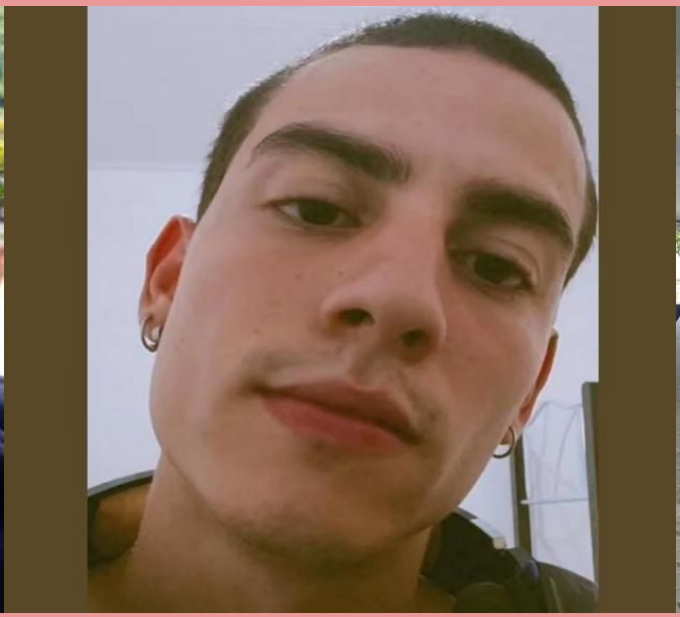

CPU[|| 2.0%]
Mem[|||||||||||| 13/123MB]
Swp[0/109MB]
Tasks: 16 total, 1 running
Load average: 0.37 0.12 0.04
Uptime: 00:00:50
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
3692 per 15 0 2424 1204 980 R 2.0 1.0 0:00.24 htop
1 root 16 0 2952 1852 532 S 0.0 1.5 0:00.77 /sbin/init
2236 root 20 -4 2316 728 472 S 0.0 0.6 0:01.06 /sbin/udevd --daem
3224 dhcp 18 -2 2412 552 244 S 0.0 0.4 0:00.00 dhclient3 -e IF_ME
3488 root 18 0 1692 516 448 S 0.0 0.4 0:00.00 /sbin/getty 38400
3491 root 18 0 1696 520 448 S 0.0 0.4 0:00.01 /sbin/getty 38400
3497 root 18 0 1696 516 448 S 0.0 0.4 0:00.00 /sbin/getty 38400
3500 root 18 0 1692 516 448 S 0.0 0.4 0:00.00 /sbin/getty 38400
3501 root 16 0 2772 1196 936 S 0.0 0.9 0:00.04 /bin/login --
3504 root 18 0 1696 516 448 S 0.0 0.4 0:00.00 /sbin/getty 38400
3539 syslog 15 0 1916 704 564 S 0.0 0.6 0:00.12 /sbin/syslogd -u s
3561 root 18 0 1840 536 444 S 0.0 0.4 0:00.79 /bin/dd bs 1 if /p
3563 klog 18 0 2472 1376 408 S 0.0 1.1 0:00.37 /sbin/klogd -P /va
3590 daemon 25 0 1960 428 308 S 0.0 0.3 0:00.00 /usr/sbin/atd
3604 root 18 0 2336 792 632 S 0.0 0.6 0:00.00 /usr/sbin/cron
3645 per 15 0 5524 2924 1428 S 0.0 2.3 0:00.45 -bash
PROCESSOS
F1Help F2Setup F3SearchF4InvertF5Tree F6SortByF7Nice -F8Nice +F9Kill F10Quit

GRUPO IF (2022) { COUT << "FAZ O L" << ENDL; }

EMANUEL BRITO



GABRIEL 69 OPPS 39

LUCAS OLIVEIRA

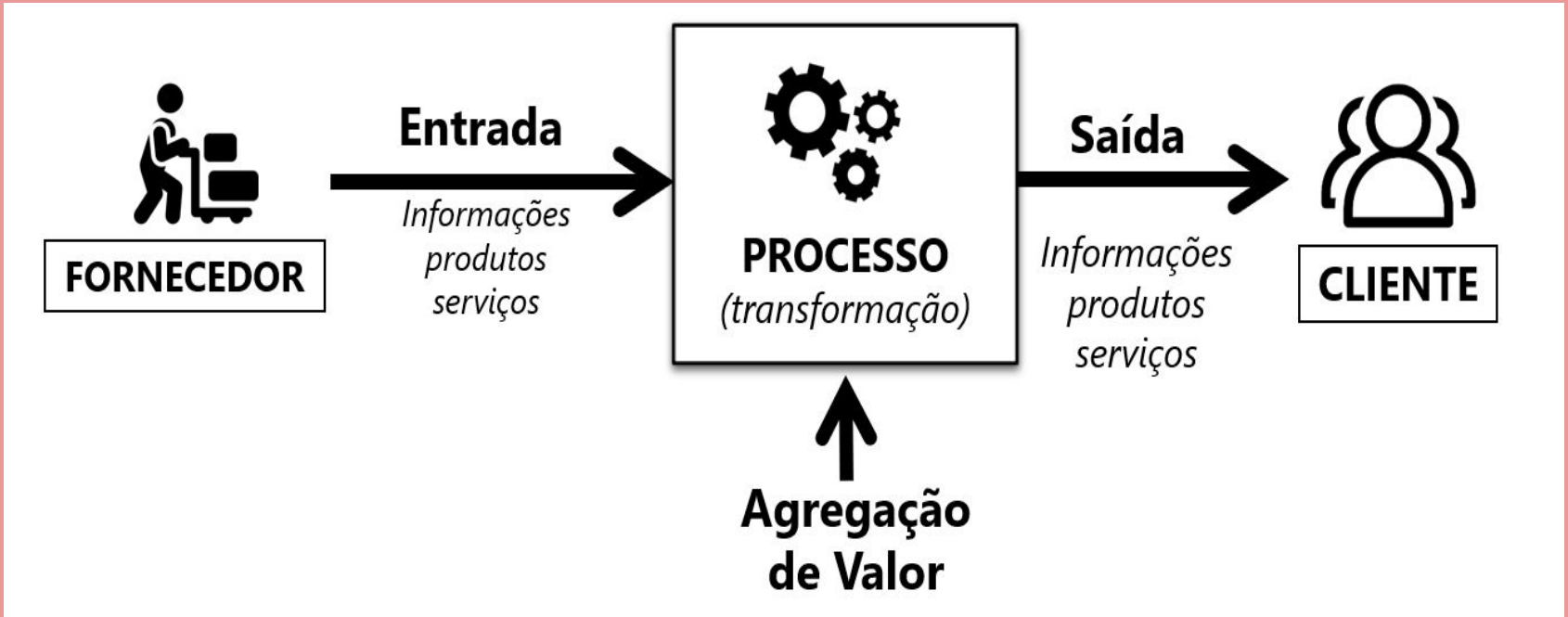
O que é um processo?

- Um sistema de computação tem mais atividades a executar que o número de processadores disponíveis. Assim foi necessário criar métodos para multiplexar o(s) processador(es) da máquina entre as atividades presentes. Além disso, diferentes tarefas possuem necessidades distintas de processamento, com isso, estratégias foram definidas para que cada tarefa receba uma quantidade de processamento que atenda suas necessidades.

- Computadores geralmente realizam diversas tarefas simultaneamente. Porém, um processador convencional somente trata um fluxo de instruções por vez, até mesmo computadores multicores possuem mais atividades a executar que o número de processadores.
- Tarefa é a execução de um fluxo sequencial de instruções, construído para atender uma finalidade específica.
- Um processo é uma forma de implementar o conceito de tarefa.
- Processos são definidos como sendo uma tarefa com seus respectivos recursos, como arquivos abertos e canais de comunicação, em uma área de memória delimitada e isolada das demais. Ou seja, um processo seria uma espécie de "cápsula" isolada de execução, contendo uma tarefa e seus recursos.

- Os sistemas operacionais mais antigos, até meados dos anos 80, suportavam somente um fluxo de execução em cada processo. Assim, as unidades de execução (tarefa) e de recursos (processo) se confundiam. No entanto, quase todos os sistemas operacionais atuais suportam e existência de mais de uma tarefa em cada processo, como é o caso do Linux, Windows, iOS e os sistemas UNIX mais recentes.
- Atualmente, o processo deve ser visto como uma unidade de contexto, ou seja, um contêiner de recursos utilizados por uma ou mais tarefas para sua execução: áreas de memória (código, dados, pilha), informações de contexto e descritores de recursos do núcleo (arquivos abertos, conexões de rede, etc). Um processo pode então conter várias tarefas, que compartilham esses recursos. Os processos são isolados entre si pelos mecanismos de proteção providos pelo hardware (isolamento de áreas de memória, níveis de operação e chamadas de sistema), impedindo que uma tarefa do processo pa acesse um recurso atribuído ao processo pb

➔ Programa x Processo/Tarefa.



Criação de Processos

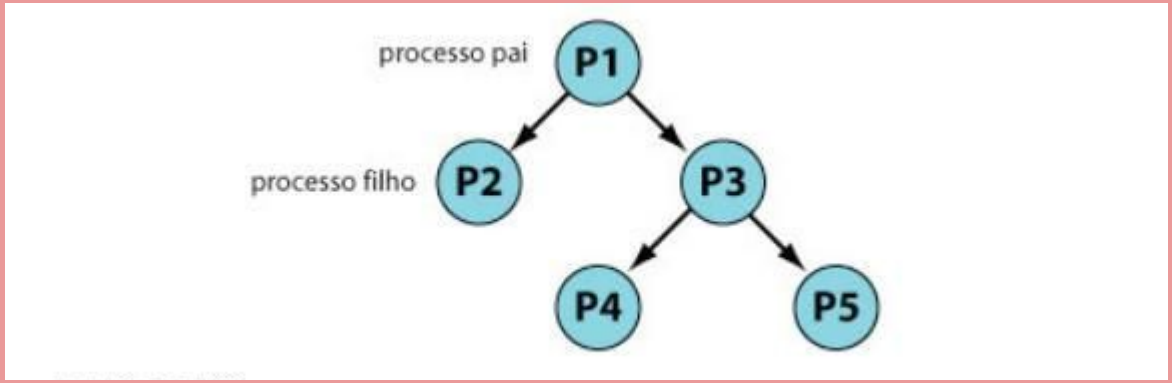
- Quatro eventos principais fazem com que os processos sejam criados
  - Inicialização do sistema.
  - Execução de uma chamada de sistema de criação de processo por um processo em execução.
  - Solicitação de um usuário para criar um novo recurso.
  - Início de uma tarefa em lote.

Término de Processos

- Um hora tudo acaba...
- A maioria dos processos termina por terem realizado o seu trabalho. O compilador executa uma chamada para dizer ao SO que ele terminou o processo.
- Mas são quatro términos possíveis para um processo:
  - Saída normal (Voluntário).
  - Erro fatal (Involuntário).
  - Saída por erro (Voluntário).
  - Morto por outro processo (Involuntário).

Hierarquia de Processos

- Um processo pai pode gerar filhos, e filhos podem gerar mais processos, e eles estão associados de certa maneira. No UNIX, processos que possuem essa hierarquia formam juntos um grupo de software e individualmente podem responder/ignorar/assumir a um sinal recebido. No UNIX é criado uma árvore de processo. O Windows não possui esse conceito de hierarquia de processos.



Estados de Processos

- Processos são entidades independentes, porém, um processo pode gerar alguma saída que outro processo use como entrada. Então, um processo pode depender de outro que ainda não foi totalmente executado, então esse processo deve ser bloqueado até que a entrada esteja disponível.
- O SO pode também decidir alocar a CPU para outro processo por um tempo. Podemos ter um diagrama de estados de processos

1. O processo é bloqueado aguardando uma entrada
2. O escalonador seleciona outro processo
3. O escalonador seleciona esse processo
4. A entrada torna-se disponível

➔ O escalonador decide qual processo deve ser executado, quando e por quanto tempo. Ele é implementado usando algoritmos de fila de prioridade etc. Podemos pensar nele como o nível mais baixo dos SOs.