COPPE UFRJ

Processos

Pedro Cruz

EEL770 – Sistemas Operacionais

Datas importantes

GTA / UFRJ

- 02 de Abril
  - Proposta de trabalho
- 09 de Abril
  - Confirmação de proposta
- 07 de Maio
  - Primeira apresentação do trabalho
- 09 de Maio
  - Entrega da primeira lista
- 14 de Maio
  - Primeira prova

Aviso

GTA / UFRJ

- O que vamos falar vale para sistemas de processador/ núcleo único
  - Vale também para cada processador/núcleo de um sistema com múltiplos processadores/núcleos

Processo

GTA / UFRJ

- Abstração para um programa sendo executado
  - Espaço de endereçamento - Contexto
    - Executável
      - Código
    - Dados
      - Informações usadas
    - Pilha
      - Armazenamento local
      - Passagem de parâmetros

Processo

GTA / UFRJ

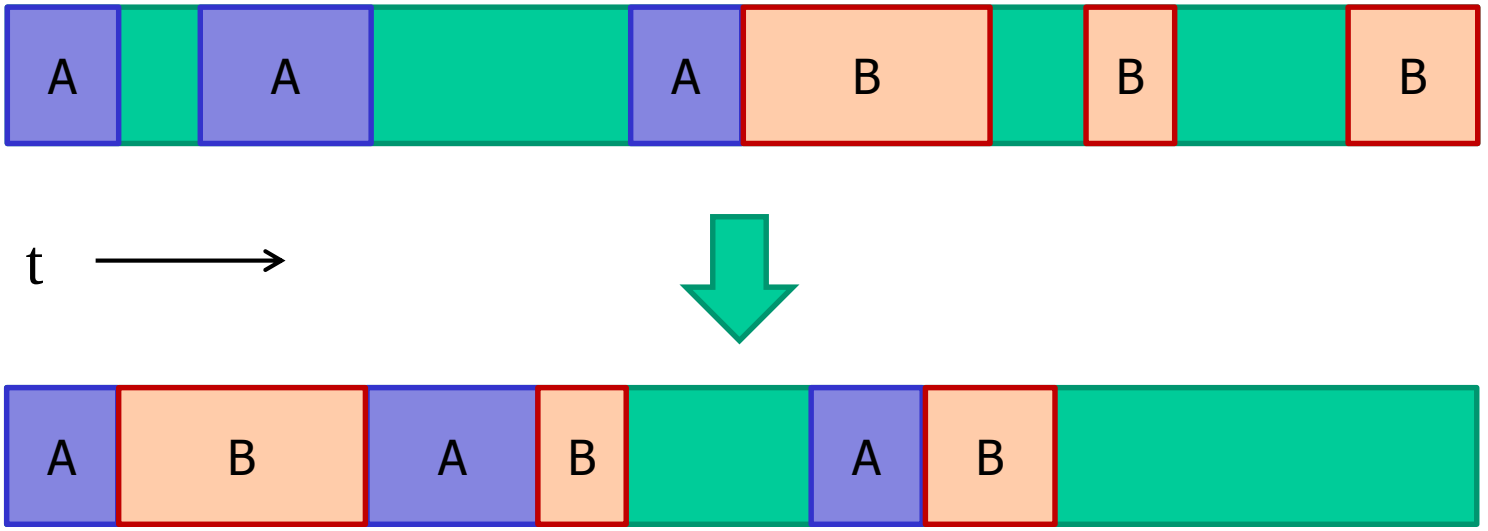
- Duas instâncias de um mesmo programa são dois processos diferentes
  - Cada um tem um espaço de endereçamento:
    - Pilha
    - Dados
    - Código*

* Em alguns casos, otimizações são feitas e o código é o mesmo, com contadores de programa diferentes

Subutilização do processador

GTA / UFRJ

- Processador é muito mais rápido que outros componentes
  - Processador fica ocioso aguardando os outros componentes
    - Ideia: executar outros processos no tempo ocioso



Pseudoparalelismo

GTA / UFRJ

- Diversos processos executados em fatias de tempo
  - Tempo do processador é subdividido e distribuído entre processos
    - Centenas ou dezenas de ms para cada processo
      - (milhões de ciclos de clock)
  - Extrapolação da ideia de reduzir o tempo ocioso
    - Melhorar a experiência do usuário, com a ilusão de paralelismo

Escalonador

GTA / UFRJ

- Define as fatias de tempo para cada processo

- Troca os contextos
  - Registradores são carregados com os valores do processo que vai ser executado
    - Contador de programa
    - Apontador de pilha
    - *Flags* e ambiente

Gerenciamento de processos

GTA / UFRJ

- Criação de processos
- Término de processos
- Hierarquia de processos
- Estados de processos

Criação de processos

GTA / UFRJ

- Processos são criados
  - Na inicialização do sistema
  - A pedido do usuário
  - A partir de um processo em execução
  - A partir de uma tarefa em lote

- Processos de primeiro plano
- Processos de segundo plano
  - *Daemons*

Pais & filhos

GTA / UFRJ

- Processo pai cria um processo filho através de chamada de sistema
  - Processo filho é uma cópia do processo pai
  - Processo pai possui "ponteiro" para filho
  - Processo filho possui próprio espaço de endereçamento
    - Exceção: *copy-on-write*
      - Espaço de memória é compartilhado, mas modificações são copiadas para área privada de memória
  - Processos podem compartilhar recursos
    - Arquivos abertos
    - Permissões

Procedimento de criação de um processo

1. Definir um identificador de processo
2. Alocar espaço de memória para o processo
3. Criar uma entrada na tabela de processos
4. Definir o estado do processo
5. Criar (ou expandir) as estruturas de dados relativas

Término de processos

GTA / UFRJ

- Formas de término
  - Voluntária
    - Saída normal
    - Saída por erro
  - Involuntária
    - Erro fatal
    - Morte por outro processo

ATENÇÃO!!!!

GTA / UFRJ

- O Tanenbaum em português troca os erros
  - Erro fatal é confundido com saída por erro

Saída normal

GTA / UFRJ

- Programa termina execução e realiza chamada de sistema para informar o término
  - Libera recursos alocados
    - Memória
    - Arquivos abertos
    - Permissões de uso exclusivo
- Chamadas para saída
  - exit
    - UNIX
  - ExitProcess
    - Windows

Saída por erro

GTA / UFRJ

- Erro interno do processo afeta seu funcionamento
  - Avisa ao usuário
  - Termina
- Processos podem tentar corrigir o erro ao invés de fechar
  - Aviso na Interface Gráfica de Usuário (*Graphic User Interface* – GUI)

Erro fatal

GTA / UFRJ

- Processo é encerrado pelo operacional
  - Instrução ilegal
  - Acesso ilegal

- Operacional pode deixar o processo lidar com o erro
  - Operacional envia um sinal ao processo quando há erro
  - Depende do erro

Lista de erros fatais

GTA / UFRJ

- Vários tipos de erros fatais
  - Tempo excedido
  - Memória não disponível
  - Erro de proteção
  - Erro Aritmético
  - Espera excedida (*Time overrun*)
  - Falha de I/O
  - Instrução inválida
  - Instrução sem privilégios
  - Mal uso de dados

Morte por outro processo

GTA / UFRJ

- Outro processo realiza chamada ao operacional para terminar o processo atual
  - kill
    - UNIX
- Processo precisa de autorização para matar outro
  - Término do pai geralmente inicia término dos filhos

Adendo – Visão do programador

- *Exception* vs *Error*
  - Linguagens diferenciam entre os dois
    - Java
    - Haskell
    - C++
      - Mais ou menos
  - Exceção é um erro lógico interno
    - Frequentemente tratável
  - Erro é um problema que não é tolerado pelo operacional
    - Aplicação nem sempre pode tentar resolver
    - Pode ser considerado problema de segurança

Hierarquia de processos

GTA / UFRJ

- UNIX
  - Pais & filhos formam um grupo de processos
    - Processos pai repassam sinais aos filhos
      - Terminação do pai é avisada aos filhos
- Windows
  - Pais possuem "ponteiro" para filhos (*handle*)
    - Pais podem passar o "ponteiro" para outros processos
      - Quebra de hierarquia

Estados de processos

GTA / UFRJ

- Novo
  - Processo é inicializado
- Pronto
  - Processo está aguardando sua vez de ser executado
- Bloqueado
  - Processo está aguardando algum evento
    - Procedimento de I/O
- Executando
  - Processo está em execução
- Terminado
  - Processo terminou

Esta lista não é exaustiva!

Estados de processos

GTA / UFRJ

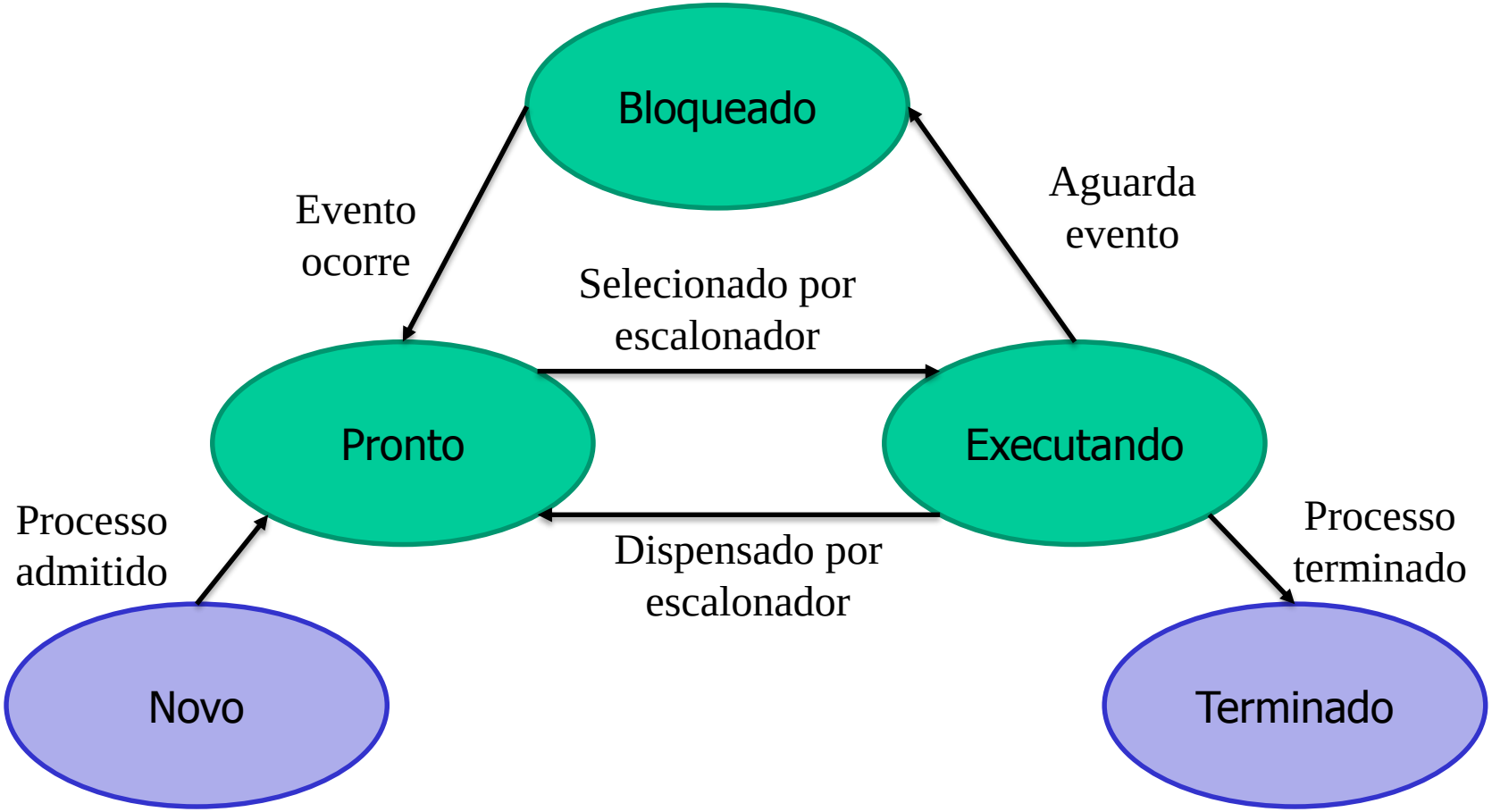


Tabela de processos – gerenciamento de processos

- Registros
- Contador de programa
- Palavra de estado do programa (PSW)
  - Conjunto de *flags*
- Ponteiro para a pilha
- Estado do processo
- Parâmetros de escalonamento
- Id do processo
- Prioridade
- Processo pai
- Grupo de processos
- Sinais
- Tempo de início
- Tempo de CPU
- Tempo de CPU de filhos

Tabela de processos – gerenciamento de memória

- Informações sobre segmento de código
- Informações sobre o segmento de dados
- Informações sobre o segmento de pilha

Tabela de processos – gerenciamento de arquivos

- Diretório-raiz
- Diretório corrente
- Descritores de arquivo
  - Ponteiros para *stdin*, *stdout*, *stderr*
- Id do usuário
- Id do grupo

Processos suspensos

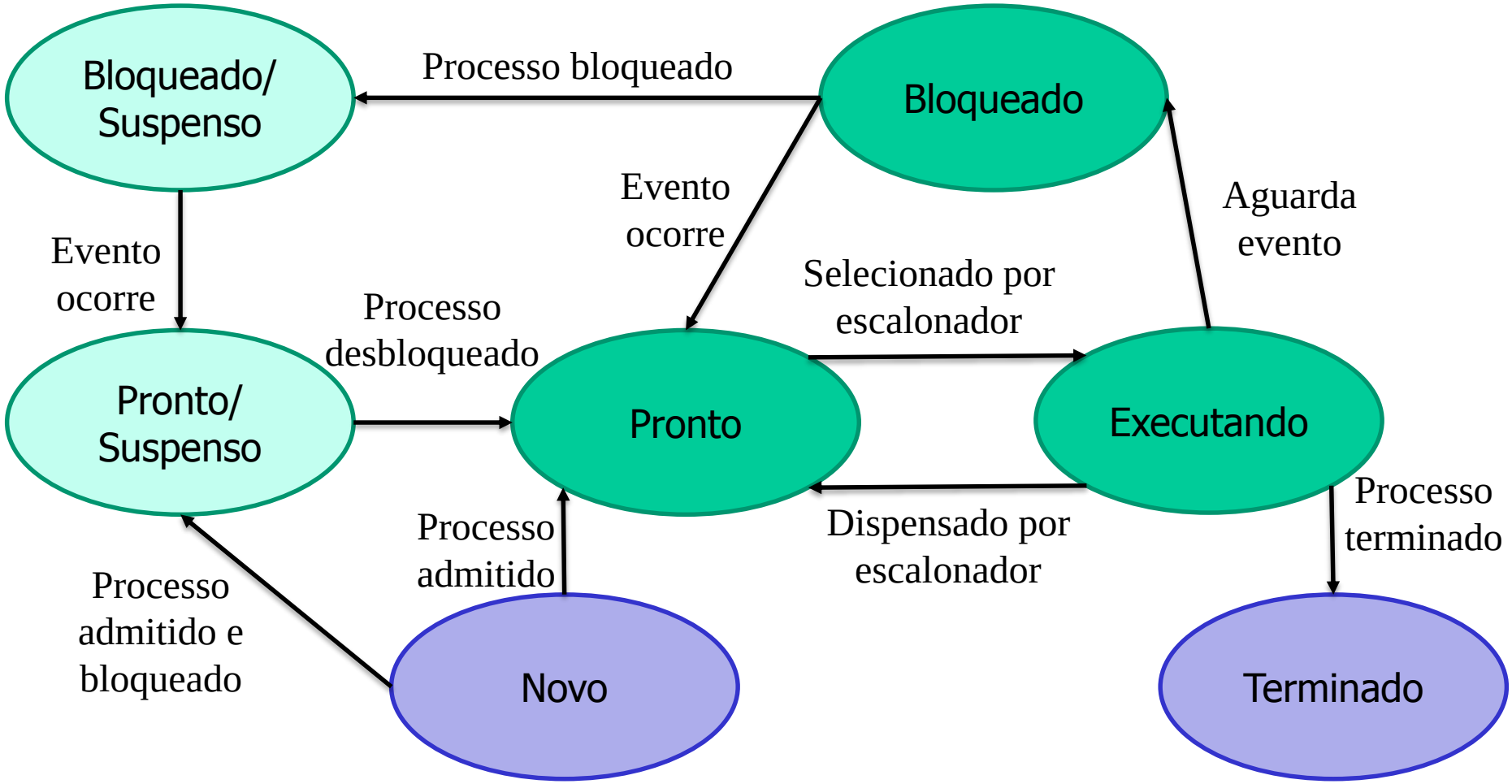
GTA / UFRJ

- Tipicamente:
  - Vários processos podem estar aguardando operações de I/O
    - Bloqueados
      - Subutilização da CPU
  - Memória principal não comporta processos suficientes para encher a CPU de trabalho
  - I/O de disco é mais rápido que outras operações de I/O

- Processos podem ser guardados na memória secundária
  - Suspensos
- Intervalos de CPU estimulam a troca de processos entre as memórias

Estados de processos com suspensão

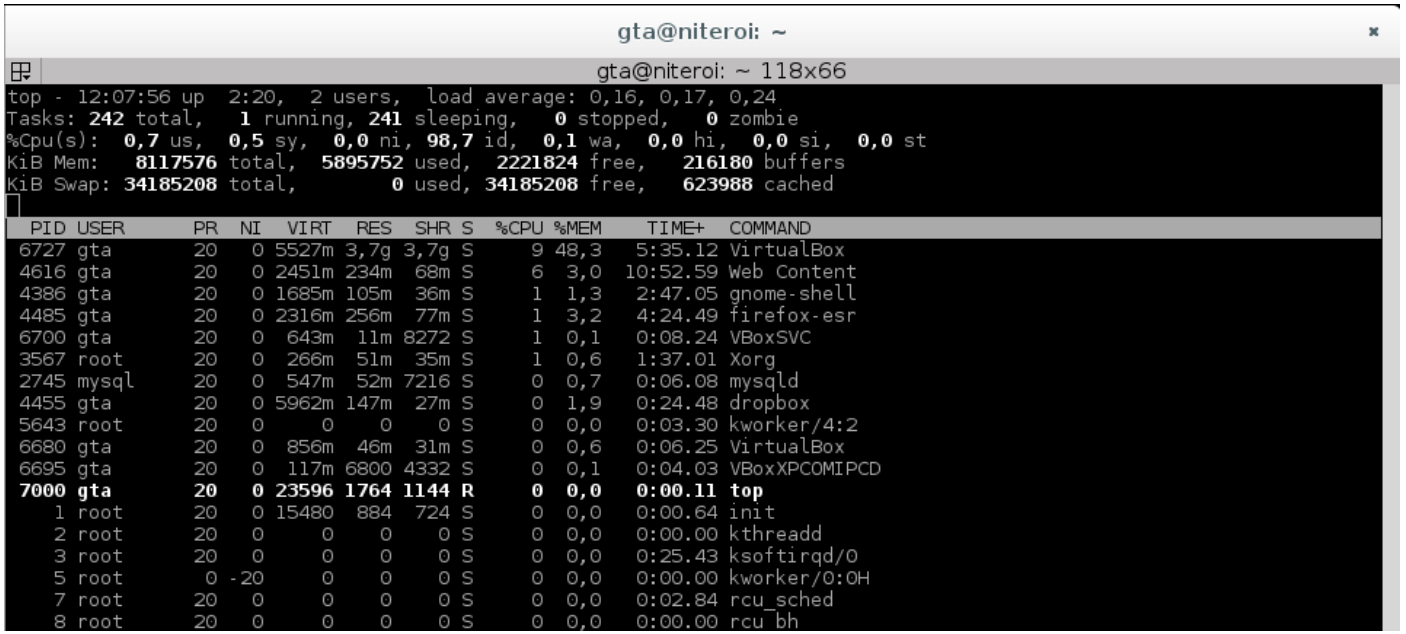
GTA / UFRJ



Top

GTA / UFRJ

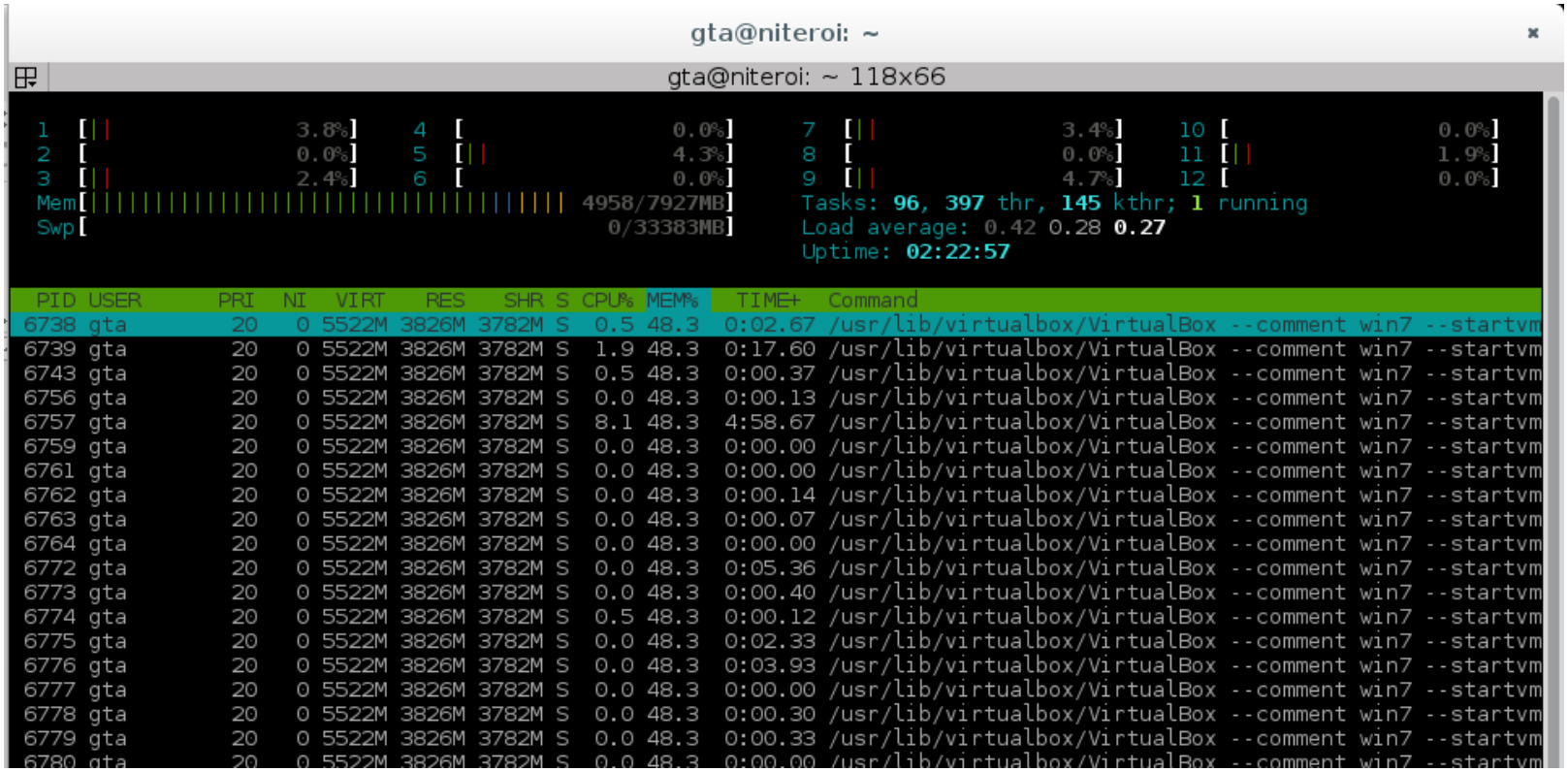
- Comando do Linux para gerenciar processos



Htop

GTA / UFRJ

- Comando do Linux para gerenciar processos



ulimit

GTA / UFRJ

- Comando do Linux para saber os limites de recursos por processo

```
gta@niteroi:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 63293
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 63293
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
gta@niteroi:~$
```

COPPE UFRJ

Processos

Pedro Cruz

EEL770 – Sistemas Operacionais