



UNIVERSIDADE ESTADUAL DO PIAUÍ
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ALINNE GRAZIELLE MESQUITA DE FARIAS
DAVI LIMA DA CRUZ
VITORIA NERIS REBELO VERAS

SISTEMAS OPERACIONAIS: ANDROID E LINUX

PARNAÍBA - PI

2021

RESUMO

Esse trabalho tem como objetivo descrever sobre os seguintes sistemas operacionais: *Android* e *Linux*. Abrangendo tópicos como a arquitetura desses sistemas, o gerenciamento de memória e do processador.

Palavras-chaves: Sistemas Operacionais. *Android*. *Linux*. Memória. Processador.

1. SISTEMAS OPERACIONAIS

Rápida introdução sobre SO's...

2. ANDROID

2.1 Arquitetura do Sistema (NERIS)

2.2 Gerência de Memória (CRUZ)

2.3 Gerência do Processador

A camada mais baixa da arquitetura *kernel Linux* é responsável pelo gerenciamento de processos e *threads*, além da memória, arquivos e pastas, redes, *drives* e dispositivos de energia. No sistema android, todos os componentes de uma aplicação são executados em um mesmo processo e *threads*. Quando a memória fica sobrecarregada, ou quando o processo faz restrição de memória, é quando ele se divide em cinco estados diferentes:

Foreground Process - nesse *status*, o aplicativo que está sendo utilizado, está sempre em primeiro plano.

Visible Process - nesse *status*, o processo não está em primeiro plano porém, está sendo utilizado e permite acesso, como exemplo: o acesso a barra de notificações.

Service Process - o processo de serviço não está vinculado a nenhum aplicativo visível em tela porém, está sendo escutado em segundo plano, por exemplo no caso em que o usuário dá o *play* em uma música e volta a utilizar algum aplicativo. O aplicativo de música será tratado como *service process*.

Background Process - os processos que estão em background podem ser considerados em estado de pausa, esses processos não tem nenhum impacto na experiência de utilizar o telefone.

Empty Process - o processo *empty* é um processo vazio. E é utilizado para poder abrir um aplicativo de forma mais rápida, da próxima que fôr inicializado.

2.3.1 Comunicação entre Processos

O *Android* oferece um mecanismo para comunicação entre processos (IPC) usando chamadas de procedimento remoto (RPCs), em que um método é chamado por uma atividade ou outro componente de aplicativo, mas é executado remotamente (em outro processo), com qualquer resultado retornado de volta ao autor da chamada. (Developers, 2021).

Pelo fato da memória ser segmentada e paginada, um processo não pode acessar diretamente a memória de outro processo. Por esse motivo foram criados os mecanismos de comunicação entre processos. Bem como foi dito acima, os sistemas *Linux* tradicionais utilizam o padrão IPC para essa comunicação, já no *Android*, esse mecanismo é chamado de *Binder*. “Os fichários são a base da arquitetura do *Android*. Ele abstrai os detalhes de baixo nível do IPC do desenvolvedor, permitindo que os aplicativos se comuniquem facilmente com o *System Server* e outros componentes de serviço remoto.” (Proandroiddev, 2020).

2.3.2 Algoritmo de Escalonamento de Processos

O conjunto de regras utilizadas para determinar quando e qual processo será executado, é conhecido como política de escalonamento. O sistema operacional *Android* divide os processos em três grandes classes, processos interativos, processo de *batch* e processos em tempo real. Em cada classe o processo ainda pode ser subdividido em *i/o-bound* ou *cpu-*

bound, de acordo com a proporção de tempo que fica esperando por operações de entrada e saída, ou utilizando o processador. O escalonador do *Android* não distingue processos interativos de processos *batch*, diferenciando-os apenas, os processos em tempo real.

O escalonador do *Android* é baseado em *time-sharing*, ou seja, o tempo do processador é dividido em fatias de tempo, chamadas de *quantum*, onde os processos são alocados, se durante a execução de um processo o *quantum* é esgotado, no processo é selecionado por execução, havendo uma troca de contexto entre os procedimentos. É completamente transparente ao processo e baseia-se em interrupção de tempo. Esse comportamento confere ao *android* um escalonamento do tipo preemptivo. Outra característica do escalonador do sistema operacional é a existência de prioridades dinâmicas, esse processo monitora o comportamento de um processo, visando equalizar o uso do processador entre os processos que ocupam o mesmo, durante um período de tempo considerado longo, tem sua prioridade reduzida, já aqueles processos que estão a muito tempo sem executar, recebem um aumento de prioridade. Sendo então beneficiados nas futuras operações de escalonamento.

3. LINUX

3.1 Arquitetura do Sistema (NERIS)

3.2 Gerência de Memória (CRUZ)

3.3 Gerência do Processador (ALINNE)

4. CONCLUSÃO

5. REFERÊNCIAS

<https://developer.android.com/guide/components/processes-and-threads?hl=pt-br>

<https://proandroiddev.com/binders-in-android-part-1-e875daeb762f>