

Trabalho Prático de Sistemas Operacionais I

Prof. Dr. Francisco José Mônaco

August 12, 2016

1 Definição

Neste trabalho você deverá criar o esqueleto de um kernel monolítico de sistema operacional compatível com o padrão POSIX para a arquitetura Intel IA-32 utilizando C e Assembly x86-32.

2 Requisitos

- **Bootloader:** você deverá escrever o código de boot usando o GRUB para inicialização e seguindo a especificação de multiboot GNU 0.6.96
- **Driver de vídeo:** utilizando E/S mapeada em memória, escreva um pequeno driver para o padrão VGA - em modo texto - que habilite seu kernel a escrever mensagens na tela.
- **Biblioteca C:** atendo-se ao padrão POSIX você deverá escrever as seguintes bibliotecas:
 1. **stdarg.h**
 2. **stdbool.h**
 3. **stddef.h:** defina **NULL** e **size_t**
 4. **stdint.h**
 5. **stdio.h:** defina somente as funções putchar, puts e printf. Sua printf deverá ser capaz de imprimir variáveis do tipo char, string, e inteiros na base binária, decimal e hexadecimal.
 6. **stdlib.h:** defina somente a função abort
 7. **string.h:** defina somente strlen, strcpy, memcpy e strcmp.
 8. **arch/registers.h:** defina uma função capaz de capturar o valor dos registradores cs, ds e ss.
- **Kernel:** esta é a parte principal do trabalho. Escreva o kernel do seu sistema operacional fazendo com que ele inicialize corretamente o driver de vídeo. Além disso, seu kernel deverá imprimir o valor dos registradores e uma mensagem de inicialização.

3 Ferramentas

Para a execução deste trabalho você deverá utilizar as seguintes ferramentas: (1) o Gnu Assembler *gas*, (2) o Gnu Linker *ld*, (3) *gcc*, (4) *Makefile* e (5) o emulador *qemu*.

4 Restrições

Atente-se para as restrições abaixo e siga-as atentamente. Note que elas são impostas para que o desenvolvimento do seu trabalho não se torne penoso ou propenso a erros e *bugs*.

1. Você não poderá, sob hipótese alguma, utilizar alguma biblioteca C disponível (*glibc*, *newlib*, *ulibc*, etc...). Faça a sua própria biblioteca C padrão seguindo as normas estabelecidas no POSIX.
2. Caso esteja em dúvida sobre se alguma ferramenta específica poderá ser utilizada, pergunte ao monitor da disciplina.
3. Minimize, ao máximo, a quantidade de código em Assembly necessário ao desenvolvimento do seu projeto. Além disso, encapsule e isole todo código dependente de arquitetura.
4. É terminantemente vedado a utilização de outras linguagem como Pascal, C++, etc...

5 Materiais e Guias

Segue algumas fontes que poderão servir de ajuda ao trabalho. É fortemente encorajado o uso pleno desses materiais durante o desenvolvimento do trabalho, respeitando sempre os direitos autorais e regras de plágio.

- Uma plataforma para ensino e treinamento em desenvolvimento de Sistemas Operacionais. Renê de Souza Pinto. 2012. Disponível em: <http://goo.gl/Ywwss8>.
- http://wiki.osdev.org/Main_Page. Uma wiki repleta de artigos, tutoriais, manuais e referências sobre a criação de sistemas operacionais.
- <https://github.com/zavg/linux-0.01>. A primeira versão do kernel do Linux.
- <https://github.com/minix3/minix>. O código-fonte do Minix, criado por Tanenbaum para o ensino de Sistemas Operacionais.
- <http://goo.gl/9exoNX>. O *Instruction Set Architecture (ISA)* da Intel.