

UNIVERSIDADE ESTADUAL DE PONTA GROSSA
SETOR DE CIÊNCIAS AGRÁRIAS E DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
ENGENHARIA DE COMPUTAÇÃO

MATEUS FELIPE DA SILVA JUNGES

SISTEMAS OPERACIONAIS – ATIVIDADES

PONTA GROSSA

2018

1. Descrever para que serve o arquivo Makefile e o comando make.

R:

O objetivo de Makefile é definir regras de compilação para projetos de software. Tais regras são definidas em arquivo chamado Makefile. O programa make interpreta o conteúdo do Makefile e executa as regras lá definidas. Alguns Sistemas Operacionais trazem programas similares ao make, tais como gmake, nmake, tmake, etc. O programa make pode variar de um sistema a outro pois não faz parte de nenhuma normalização .

Vantagens do uso do makefile:

- Evita compilação de arquivos desnecessários;
- Automatiza tarefas rotineiras
- Pode ser usado como linguagem geral de script

2. Fazer um relatório descritivo sobre o funcionamento do bootloader do arduino ATmega.

R:

Arquivo ["Análise Descritiva – Bootloader arduino.pdf"]

3. Revisar o Apendice A (página 154), do livro de “Linguagem de Programação C”, e, em profundidade os seguintes conteúdos:

a) Argumentos – passagem por valor, p. 28:

É a forma mais comum de passagem de parâmetros. Exemplo

```
float seno (float angulo); <<Ângulo é o parâmetro, passado por valor>>
```

b) Variáveis externas p. 31 e 63.

A finalidade de declarar um identificador externamente é tornar a variável disponível para um grupo de funções em vez de apenas para uma função. As variáveis declaradas antes da função main() são definidas como identificadores globais.

c) arquivos de cabeçalho (header), p. 69;

Arquivo que permite separar certos elementos do código fonte de um programa em arquivos reutilizáveis. Geralmente contém declarações de classes, subrotinas, variáveis e outros identificadores.

d) pre-processador da linguagem, p. 74;

Essencialmente, um pré-processador é um processador de macros para uma linguagem de alto nível. O programador se comunica com o pré processador utilizando diretivas em um código fonte para facilitar a manutenção do código. As diretivas para o pré processador C podem ser reconhecidas pelo símbolo # na primeira coluna de cada linha em que ocorrem.

e) parâmetros da linha de comando, p. 95;

Na linguagem C, dá pra passar argumentos através de linha de comando para um programa quando ele inicia. A função main recebe parâmetros passados via linha de comando da seguinte forma:

```
int main(int argc, char *argv[])
```

Onde:

Argc = valor inteiro que indica a quantidade de argumentos que foram passados ao se iniciar o programa

Argv = vetor de char que contem os argumentos, um para cada string passada na linha de comando.

Argv[0]= armazena o nome do programa que foi chamado no prompt, sendo assim, argc é pelo menos igual a 1.

Os argumentos passados em linha de comando devem ser separados por um espaço em branco ou tabulação.

f) ponteiro para função, p. 98;

O endereço de uma função é acessível ao programador através de uma variável do tipo ponteiro para função. Ponteiro para funções podem ser passados como argumentos para outras funções, e a função apontada pode ser invocada a partir de seu ponteiro.

g) union, p.120;

A mesma área de memória é compartilhada pelas variáveis definidas dentro da union.

h) campos de bit, p. 121;

Além dos declaradores para membros de uma estrutura ou união, um declarador de estrutura também pode ser um número especificado de bits, chamado de “campo de bits”. O comprimento é definido fora do declarador do nome de campo por dois pontos. Um campo de bits é interpretado como um tipo integral.