UNIVERSIDADE DE SÃO PAULO – USP
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
DEPARTAMENTO DE SISTEMAS DE COMPUTAÇÃO

Relatório Sistema Aeronave

Adams Vietro Codignotto da Silva - 6791943
Sabrina Faceroli Tridico - 9066452
Lucas Bortolini Fronza - 8124184

São Carlos
2017

Tabela de Conteúdo

1 Introdução

Neste projeto foi utilizado a linguagem C/C++, desenvolvido e testado apenas em ambientes Linux e Mac, e seu funcionamento em Windows não é garantido. Foram utilizadas bibliotecas padrões nativas do gcc, então não é necessário de nenhum pacote adicional.

2 Socket

Para a implementação do socket, foram utilizadas principalmente as bibliotecas *sys/socket.h* e *netdb.h*. Foi escolhida uma abordagem de cliente/servidor utilizando conexão UDP com chamada bloqueante, pois foi esta que mais tivemos documentações disponíveis online. Outras funções exclusivas do sistema Linux foram utilizadas, como *getaddrinfo* e *freeaddrinfo*.
A estrutura do cliente e do servidor são parecidas, ambos construtores recebem um endereço e uma porta para comunicar, e recebem as mesmas propriedades de conexão (como protocolo e tipo de socket), porém o servidor é associado a um único endereço, criando uma chamada bloqueante.
As funções *send* e *recv* foram utilizadas para enviar e receber mensagens pelo endereço/porta definidos nos construtores.

3 Sensores

Para este projeto decidimos pela criação dos sensores reais *Altitude*, *Temperatura*, *Distância*, *Velocidade do avião*, *Direção do vento* e *Quantidade de passageiros* e dos sensores virtuais *Peso*, *Rota*, *Aliens* e *Nível da zoeira*. Todos os sensores virtuais utilizam 3 valores obtidos pelos sensores reais como entrada.
Cada sensor real possui um canal de comunicação com o *gerenciador* e o mesmo é responsável por receber todos esses dados, calcular os dados dos sensores virtuais e mostrar todas essas informações para o usuário.

4 Manual de Usuário

Disponibilizamos um *Makefile* e, portanto, basta utilizar o comando *make all* para compilar e o comando *make run* para executar o programa. O *make run* iniciará o *gerenciador*, o qual irá iniciar todos os sensores.
Os sensores rodam separadamente como um programa, enviando informações para o gerenciador. Ao iniciar o gerenciador, todos os sensores são inicializados. Se o gerenciador é encerrado, uma função cuida de encerrar o processo de cada sensor. Para encerrar o gerenciador, o *CTRL+C* é necessário, uma vez que tudo roda em um loop infinito.