

Universidad Autónoma de Querétaro

Facultad de Ingeniería

Programación

Bot

Mtro. José de Jesús de Santana Ramirez.

Integrantes:

García Alvarado Karla Ximena

Mandujano Gómez Hugo Alberto

Ramirez Tejeida Daira Lissette

Romero Lugo Aaron

Bot

19/02/2019

1 Introducción

Un bot (aféresis de robot) es un programa informático que efectúa automáticamente tareas repetitivas a través de Internet, cuya realización por parte de una persona sería imposible o muy tediosa.

Algunos ejemplos de bots son los rastreadores web de los motores de búsqueda de Internet, que recorren los sitios web de forma automática y recopilan información de los mismos de manera mucho más rápida y efectiva de lo que lo haría una persona. Los bots "buenos" cumplen los estándares de exclusión de robots, que los operadores de servidores pueden usar para influir en el comportamiento de un robot dentro de unos límites.

Los bots "maliciosos" se utilizan, por ejemplo, para recopilar direcciones de correo electrónico con fines publicitarios, para hacer copias masivas no autorizadas de contenidos web o para espiar de manera sistemática las vulnerabilidades de software de los servidores con el objetivo de penetrar en ellos. En las redes sociales, los bots se utilizan para simular la interacción humana, hinchando artificialmente el número de visitas o seguidores, o automatizando respuestas para posicionar mensajes o influir en debates. Los denominados bots conversacionales son sistemas de inteligencia artificial que simulan una conversación con una persona utilizando el lenguaje natural.

Es importante distinguir que bot es una definición funcional, y no hace diferencias en cuanto a su implementación. Un bot puede estar diseñado en cualquier lenguaje de programación, funcionar en un servidor o en un cliente, o ser un agente móvil, etc. A veces son llamados Sistemas Expertos, pues muchos se especializan en una función específica.

La programación de un bot puede estar diseñada para cumplir tareas muy básicas como lo son el recordar alguna tarea o bien automatizar algún proceso, también existen bots con programación más compleja que buscan realizar actividades que conllevan toma de decisiones; estas decisiones son tomadas a partir de filtros o parámetros que el programador incluye en el código de programación. En estos días existen asistentes avanzados amigables que ayudan a manipular más fácilmente el código del bot.

2 Usos

Los principales usos de los bots son:

Rastrear información en la web (web crawler): con un script automático se capta, analiza y almacena información de un servidor web de forma automatizada y mucho más rápido de lo que lo haría una persona. Cada servidor puede tener un archivo llamado robots.txt, que contiene reglas para su rastreo. Dar respuestas rápidas: los bots de juegos o los de webs de subastas en línea ofrecen respuestas de manera mucho más rápida y efectiva que un humano. Mantener conversaciones: los bots conversacionales simulan una conversación humana al proveer respuestas automáticas a entradas hechas por el usuario.

Simular tráfico en Internet y las redes sociales: los bots se utilizan para aumentar el número de seguidores de manera artificial, para generar artificialmente visitas a sitios web, para posicionar hashtags y temas para convertirlos en trending topics, o para influir positiva o negativamente en

una conversación.

Chat en línea

Algunos bots se comunican con otros usuarios de servicios basados en Internet, mediante mensajería instantánea (MI), Internet Relay Chat (IRC) u otros interfaces web. Estos bots permiten a una persona hacer preguntas en cualquier idioma y recibir las correspondientes respuestas. Los bots a menudo sirven para realizar múltiples tareas a la vez, como por ejemplo dar información sobre el tiempo, sobre geografía, sobre deportes, hacer la conversión de monedas u otras unidades. Un uso alternativo de los bots de IRC es el de residir en un canal de chat, comentando ciertas frases escritas por los participantes (basándose en el reconocimiento de patrones). Esto se utiliza como servicio de ayuda a los nuevos usuarios, o para censurar el lenguaje soez.

Videojuegos

En el ambiente de los videojuegos, se conoce como bot a programas que son capaces de jugar por sí mismos el juego en cuestión (también conocidos como borgs). La calidad del bot en este caso viene determinada por su capacidad de vencer (y en qué términos) el videojuego. Los bots para juegos CRPG (computer role-playing games) son particularmente conocidos ya que este tipo de aplicaciones requieren una gran capacidad de estrategia para ganarlos. Es muy habitual también usar este término en los juegos disparos en primera persona, en los que sustituyen a un jugador humano cuando no hay contrincantes disponibles o en juego offline.

2.1 Dato curioso

Los bots se pueden usar para fines maliciosos a través de Internet; por ejemplo, el ataque de denegación de servicio, en el que un buen número de máquinas atacan un servidor web (véase Botnet). Otros ejemplos de uso malicioso son los bots de SPAM, que acumulan direcciones de e-mail rastreando páginas de contacto o de libros de visita, bots que ocupan todo el ancho de banda descargando sitios webs enteros, virus, gusanos, bots que incrementan artificialmente visitas a sitios web.

La herramienta más usada contra estos bots es el sistema CAPTCHA, una forma de test de Turing en el que se usa una palabra presentada en formato gráfico que el usuario debe escribir, algo muy difícil de interpretar para un bot. No obstante, este sistema no es fiable al cien por cien, puesto que puede ser esquivado por programas de reconocimiento de caracteres o agujeros de seguridad.

Los bots también se usan para comprar los mejores asientos para conciertos y eventos, con el propósito de revenderlos posteriormente. Los reventas compran automáticamente las entradas a través de bots para obtener de manera fraudulenta los mejores asientos para sí mismos, privando al público general de dichos asientos.

En los juegos en línea, se utilizan bots para desempeñar tareas repetitivas para conseguir recursos o mejorar los valores de los personajes, que de manera ordinaria requerirían mucho tiempo o esfuerzo obtener.

3 Código

Se utilizó el siguiente código para generar el bot:

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    printf("-BIENVENIDO!!!!-\n");
    char ch;
    ch=getchar();
    printf("\nCon que orientacion sexual te identificas??\n");
    printf("a.-hombre \nb.-mujer \nc.-homosexual \n");
```

```

scanf("%c", &ch);
switch(ch)
{
    case 'a': printf("Hola bro :v\n\n");
    break;
    case 'b': printf("Hola amiga ;)\n\n");
    break;
    case 'c': printf("Como estas bb? 7u7\n\n");
    break;
    default: printf("Error\n\n");
}
char n;
n=getchar();
printf("Te gustan mas los perros o los gatos? n.n\n");
printf("a.-Lomitos\nb.-Gatetes\nc.-Me gustas tu 7u7\nd.-No me gustan los
    animales ._.\n");
scanf("%c", &n);
switch(n)
{
    case 'a': printf("Ahhh perroooo... que gustos mas finos xD\n\n");
    break;
    case 'b': printf("A mi tambien me maman los michis :)\n\n");
    break;
    case 'c': printf("Hay gracias 7u7\n... pero...\nNO ERES MI TIPO U.U\n\n");
    ;
    break;
    case 'd': printf("Pues menos me gustas tu e.e\n\n");
    break;
    default: printf("Error\n\n");
}
char n1;
n1=getchar();
printf("Y que tal te va??\n");
printf("a.-He tenido un buen dia hoy :)\n b.-Hoy no es mi dia : 'c\nc.-Pues
    la vida ya me es indiferente alv\n");
scanf("%c", &n1);
if(n1=='a')
{
    char n1a;
    n1a=getchar();
    printf("Me alegra mucho ;)\nOjala tengas mas dias asi :)\n");
    printf("Y por que ha sido un buen dia? :)\n");
    printf("a.-Asi siempre son mis dias B^\nb.-Me paso algo bueno :)\nc
        .-Y por que no habria de serlo? xD\n");
    scanf("%c", &n1a);
    switch(n1a)
    {
        case 'a': printf("Jajajaja que buena actitud!!! xD\n\n");
        break;
        case 'b': printf("Que bueno bro :)\n\n");
        break;
        case 'c': printf("Ah 'inche perro :o\n\n");
        break;
        default: printf("Error\n\n");
    }
}
else
    if(n1=='b')
    {
        char n1b;
        n1b=getchar();
        printf("Entiendo tu dolor u.u\n");
        printf("Y se puede saber por que ha sido asi? :(\n");
    }

```

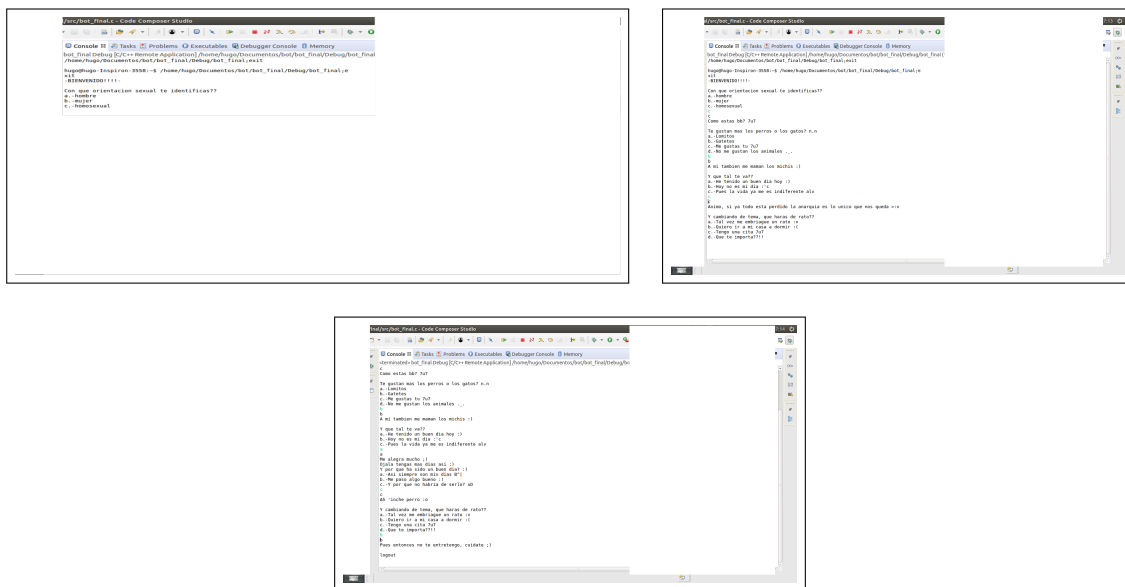
```

printf("a.-Preferiria no hablar de eso :/\nb.-Tuve un conflicto con
      una persona u.u\nc.-El mundo es una mierda >:v\nd.-Asi son mis
      dias : 'v\n");
scanf("%c", &n1b);
switch(n1b)
{
case 'a': printf("Ok bro, se entiende u.u\n\n");
break;
case 'b': printf("Lamento oir eso :c\n\n");
break;
case 'c': printf("El mundo es caotico, desconoce y ni le importaria
      ser justo\n\n");
break;
case 'd': printf("Chale, eso no suena nada bien amig@ :(\n\n");
break;
default: printf("Error\n\n");
}
}
else
    if(n1=='c')
    {
printf("Animo, si ya todo esta perdido la anarquia es lo unico
      que nos queda >:v\n\n");
    }
    else
    {
        printf("Error!!\n\n");
    }
}

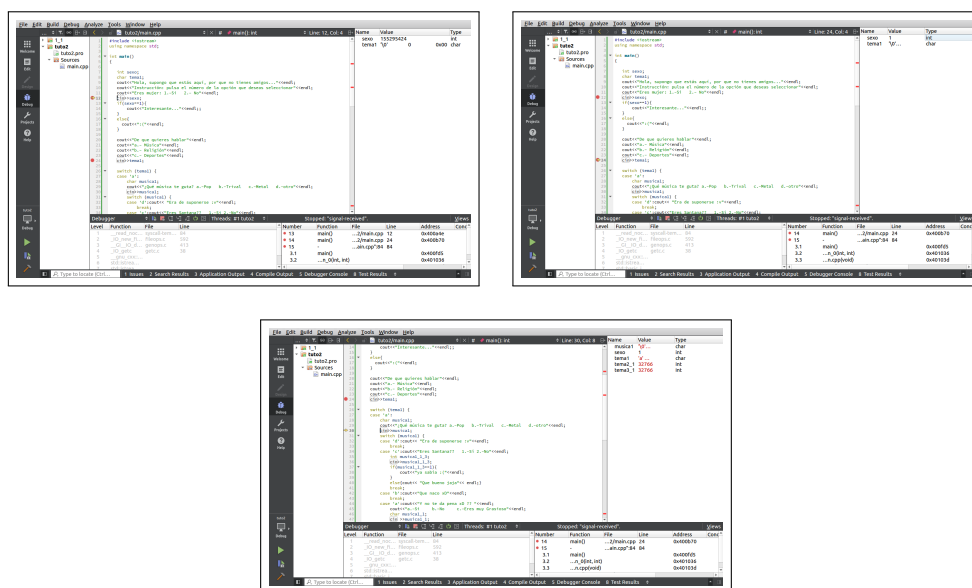
char n2;
n2=getchar();
printf("Y cambiando de tema, que haras de rato??\n");
printf("a.-Tal vez me embriague un rato :v\nb.-Quiero ir a mi casa a
      dormir :(\nc.-Tengo una cita 7u7\nd.-Que te importa??!\n");
scanf("%c", &n2);
switch(n2)
{
case 'a': printf("Pues ya es hora de que saques las caaaguaaaamaaaaaasss
      ;)\n");
      printf("Jalas o no??\n");
      char n2a;
      n2a=getchar();
      printf("a.-Jalo ;) \nb.-Nel perro >:v\nc.-Otro dia :)");
      scanf("%c", &n2a);
      switch(n2a)
      {
case 'a': printf("Pues vamos ;) \n\n");
break;
case 'b': printf("Pues entonces ya vete payaso e.e\n\n");
break;
case 'c': printf("Camara tons :v\n\n");
break;
default: printf("Error\n\n");
}
break;
case 'b': printf("Pues entonces no te entretengo, cuidate ;) \n\n");
break;
case 'c': printf("Que buen pedo :) \nQue te vaya bien amig@ ;) \n\n");
break;
default: printf("Error\n\n");
}
return 0;
}

```

3.1 Capturas de la terminal



3.2 Debug



4 Conclusión

En esta segunda práctica de laboratorio, cuyo objetivo era programar un bot personalizado con quien pudiéramos tener una conversación, me di cuenta de la importancia que tiene el ser ordenados al momento de escribir nuestros códigos, ya que, al estar tratando con diferentes variables y tipos de escritura en `c++`, ante cualquier pequeño error nuestro programa no correrá; sin embargo, si mantenemos una estructura clara y concisa será fácil identificar nuestro error, como comúnmente es el uso de llaves o la falta de estos. Nosotros escogimos utilizar la condición `switch` anidada, donde retomo el punto clave que consiste en mantener un orden conciso. A su vez, considero que pudimos aprender al respecto en esta actividad, por el simple hecho de relacionarnos un poco más con los comandos al irlos escribiendo, como bien dicen, uno aprende más conforme a la práctica.

References

<https://es.wikipedia.org/wiki/Bot>

<https://vilmanunez.com/bot-facebook/>