## Interpretación de los resultados del programa en el directorio MTX4\_0C

/\*\*\* Incluir aquí los códigos de los archivos del kernel MTX4\_0C \*\*\*/

Con el programa del directorio MTX4\_0C se encontraron los descriptores de segmento (de la GDT) que está usando el programa. El programa usa 5 descriptores de segmento, el primero es el descriptor nulo (8 bytes 0), el segundo, como se verá, es un descriptor de segmento de código (Bit Ex=1) con privilegio de kernel (Privl=00), el tercero es un descriptor de segmento de datos (Bit Ex=0) con privilegio de kernel. También, como se verá, estos dos descriptores tienen un límite de FFFFF=%FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
2^{20}-1
\]
\end{document}, es decir el límite del segmento es de %FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
2^{20}
\]
\end{document} bloques de 4096 bytes cada uno (ya que el bit Gr en los dos descriptores es igual a 1). Los dos descriptores tienen direcciones base de 0x00000000, por lo cual, ambos descriptores pueden direccionar hasta %FontSize=11
%TeXFontSize=11
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
2^{20}\times 2^{12}=2^{32}=4\,GB
\]
\end{document}, lo cual corresponde al modelo de memoria plano.

Los dos descriptores de segmento a los que se hace referencia en el párrafo anterior son:

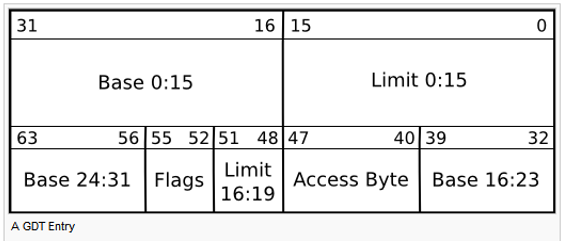
Descriptor con índice 0x08:

00 CF 9A 00 00 00 FF FF

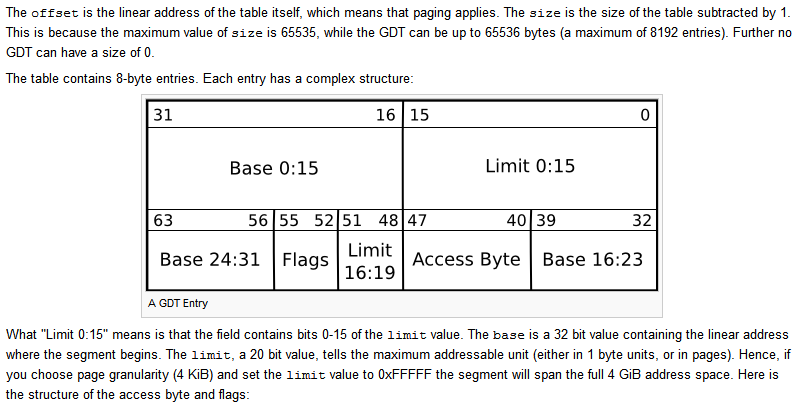
Descriptor con índice 0x10

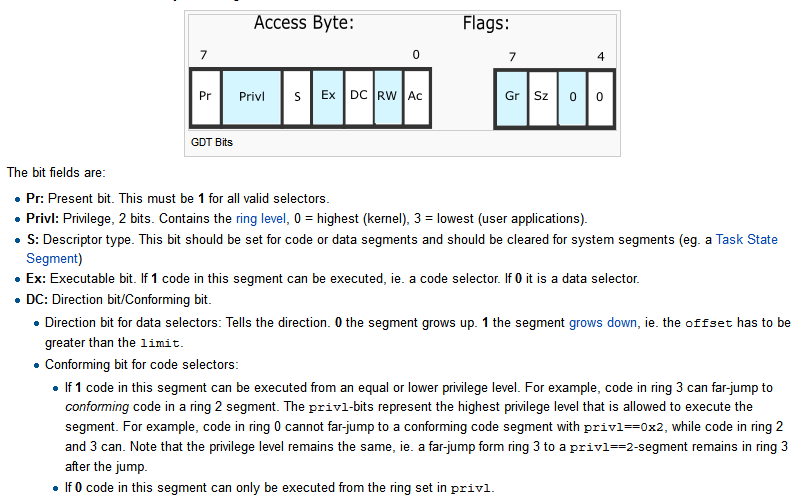
00 CF 92 00 00 00 FF FF

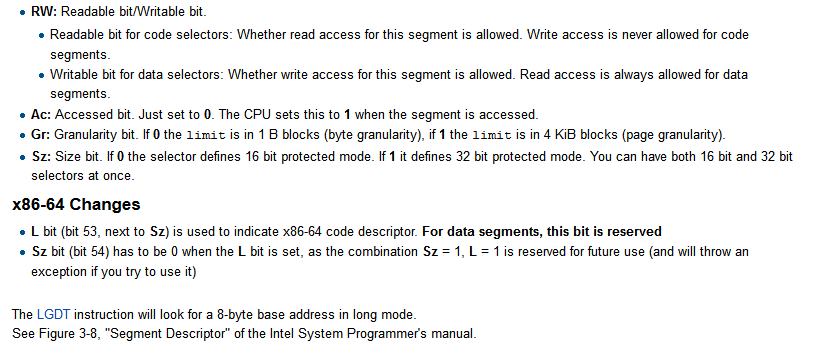
Cada uno de los descriptores contiene la siguiente información:



**Tabla 1** Contenido de un descriptor de segmento







A partir de la Tabla 1, para el descriptor con índice 0x08, tenemos que

%FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{tabular}{|l|r|r|}\hline
Base  & 00000000&00000000000000000000000000000000\\\hline
Limit & FFFFF & 11111111111111111111\\\hline
Access Byte & 9A & 10011010\\\hline
Flags & C & 1100\\\hline
\end{tabular}
\end{document}

Descriptor de segmento con índice 0x08 desglosado

El byte de acceso tiene los siguientes bits:



El bit de ejecución Ex=1 y los bits del nivel de privilegio Privl= 0 0 nos indican que se trata de un segmento de código con nivel de privilegio de modo kernel.

El nibble de Flags tiene los siguientes bits:

%FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{tabular}{|c|c|c|c|}\hline
Gr&Sz&0&0\\\hline
1 & 1 & 0 & 0\\\hline
\end{tabular}
\end{document}

el bit de granularidad Gr=1 nos indica que el límite Límit está expresando en bloques de 4 KiB (bloques de 4096 bytes), y el byte Sz=1 nos indica que el segmento es del modo protegido de 32 bits.

La base 0x00000000 y el Limit de 0xFFFFF junto con la granularidad Gr=1 nos indican que el segmento puede direccionar hasta (2^{20}-1)+1=2^{20} bloques de 4 KiB = 4096 bytes cada uno, es decir, hasta 4 GB de memoria.

A partir de la Tabla 1, para el descriptor con índice 0x10, tenemos que

%FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{tabular}{|l|r|r|}\hline
Base  & 00000000&00000000000000000000000000000000\\\hline
Limit & FFFFF & 11111111111111111111\\\hline
Access Byte & 92 & 10010010\\\hline
Flags & C & 1100\\\hline
\end{tabular}
\end{document}

Descriptor de segmento con índice 0x10 desglosado

El byte de acceso tiene los siguientes bits:



El bit de ejecución Ex=0 y los bits del nivel de privilegio Privl= 0 0 nos indican que se trata de un segmento de datos con nivel de privilegio de modo kernel.

El nibble de Flags tiene los siguientes bits:

%FontSize=12
%TeXFontSize=12
\documentclass{article}
\pagestyle{empty}
\begin{document}
\begin{tabular}{|c|c|c|c|}\hline
Gr&Sz&0&0\\\hline
1 & 1 & 0 & 0\\\hline
\end{tabular}
\end{document}

el bit de granularidad Gr=1 nos indica que el límite Límit está expresando en bloques de 4 KiB (bloques de 4096 bytes), y el byte Sz=1 nos indica que el segmento es del modo protegido de 32 bits.

La base 0x00000000 y el Limit de 0xFFFFF junto con la granularidad Gr=1 nos indican que el segmento puede direccionar hasta (2^{20}-1)+1=2^{20} bloques de 4 KiB = 4096 bytes cada uno, es decir, hasta 4 GB de memoria.