

Laboratori de telemàtica

Sessió 2

a)

El caràcter *BACKSPACE* és 0x08. Alguns terminals envien la seqüència *BackSpace*, *Space*, *BackSpace*. L'objectiu d'aquesta seqüència és esborrar el caràcter escrit anteriorment, és per això que tiren enrere un caràcter (*backspace*), escriuen un espai en blanc (caràcter 0x20) per a sobreescriure l'anterior caràcter i insereixen un nou *backspace*, per a compensar el desplaçament produït pel caràcter *space*. Altres terminals, en canvi, només envien el caràcter *backspace*.

b)

La transferència del fitxer labtel1.txt va trigar 7,2 segons aproximadament. Fent el càlcul teòric del que hauria de trigar veiem que:

$$878 \text{ bytes} \cdot \frac{10 \text{ bits}}{1 \text{ byte}} \cdot \frac{1 \text{ segon}}{1200 \text{ bits}} = 7.31 \text{ segons}$$

Lògicament esperàvem un temps semblant o superior. Tot i que la diferència pot no semblar gaire gran podem calcular el nombre de caràcters que es podrien haver transmès en aquesta diferència de temps:

$$7.31 - 7.2 \text{ segons} = 0.11 \text{ segons} \quad 0.11 \text{ segons} \cdot \frac{1200 \text{ bits}}{1 \text{ segon}} \cdot \frac{1 \text{ byte}}{10 \text{ bits}} \approx 13 \text{ bytes}$$

Veiem doncs que és molt estrany ja que la diferència de temps és de quasi 13 caràcters. La resposta és que el nostre terminal envia els salts de línia amb només un caràcter (LF), mentre que al fitxer en tenim dos (CR i LF). És comú als entorns Windows que es facin servir dos caràcters per al final de línia, mentre que als Unix només un.

Si canviem les preferències del terminal per a usar CR+LF llavors obtenim un temps de 7,28 segons, molt més proper als 7,31 ideals. La petita desviació pot ser deguda als rellotges de les UARTs, ja que els 1200 bits/seg no tenen per què ser exactes.

c)

El caràcter Xoff té com a objectiu indicar a l'altre extrem que no envii res més. Això s'utilitza quan el buffer d'entrada està proper al desbordament i així s'aconsegueix aturar la transmissió fins que el buffer s'ha buidat (s'han processat les dades).

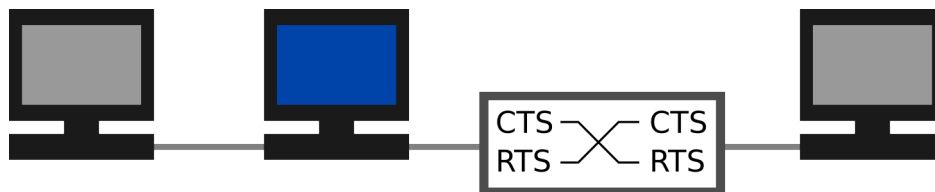
Si intentem enviar caràcters o un fitxer des d'un extrem i rebem un Xoff de l'altre, es pausa temporalment la transmissió fins que es rep un nou caràcter Xon. El que succeeix és que el terminal que envia guarda les dades que no pot trametre en un buffer i espera a rebre el Xon per reprendre la transmissió. La nostra experiència és que si escrivim una frase i enviem un Xoff des de l'altre PC la frase es talla en el moment de rebre el Xoff. En el moment en que enviem un Xon s'envia el que queda de la frase de cop (ja que estava guardat al buffer del programa).

L'efecte de les línies RTS i CTS és nul. Podem desconnectar-les i la transferència funciona sense cap problema, ja que són ignorades.

La codificació de Xon i Xoff és, respectivament, 0x11 (00010001b) i 0x13 (00010011b), podem veure-ho a l'oscil·loscopi i també a les taules ASCII.

d)

Donat que les línies CTS i RTS donen permís a l'altre extrem per a transmetre si desconnectem una d'elles l'altre extrem no podrà transmetre, ja que no tindrà permís. És similar a si enviéssim un Xoff.



En aquest esquema al desactivar la línia CTS<-RTS el que es produeix és que el PC de la dreta és capaç de transmetre però el de l'esquerra no pot transmetre, ja que la seva entrada CTS està a OFF i això significa que no té permís per a funcionar. Si ara desconnectem el cable RTS->CTS i connectem l'anterior succeeix el mateix però es canvien els papers. En el cas que no connectem cap dels dos cables el que succeeix és que ningú pot transmetre, i per tant, rebre res.

La posició del null-modem és important des del punt de vista de l'analitzador de protocols, ja que depenent de quin costat analitza ens mostra un ordinador o un altre com a DTE. En el cas d'enviar Xon o Xoff i utilitzant control de flux per hardware el que passa és que són ignorats i tractats com un altre caràcter més.

e)

Per aconseguir bloquejar el terminal receptor cal seleccionar text. Això fa que no llegeixi les dades que li envia la UART i, per tant, no es buida el buffer. Si enviem moltes dades des de l'altre extrem arriba un moment en que el buffer s'omple i el PC apaga la senyal RTS (en cas de control de flux per hardware) o bé envia el caràcter Xoff (en cas de software). De qualsevol de les dues maneres tan aviat com desbloquegem el PC aquest processa les dades pendents i puja de nou RTS o envia un Xon per a continuar rebent més dades.

Cal notar que no espera a que tot el buffer s'ompli, quan arriba a un cert llindar avisa que no vol més dades. Això ho fa per què el temps de resposta de l'altre extrem pot ser lent i trigar una mica a parar la transmissió. En la nostra experiència un cop s'avisa que no es vol més dades s'envien uns 4 o 5 caràcters de més, que són el temps de reacció que té el PC.